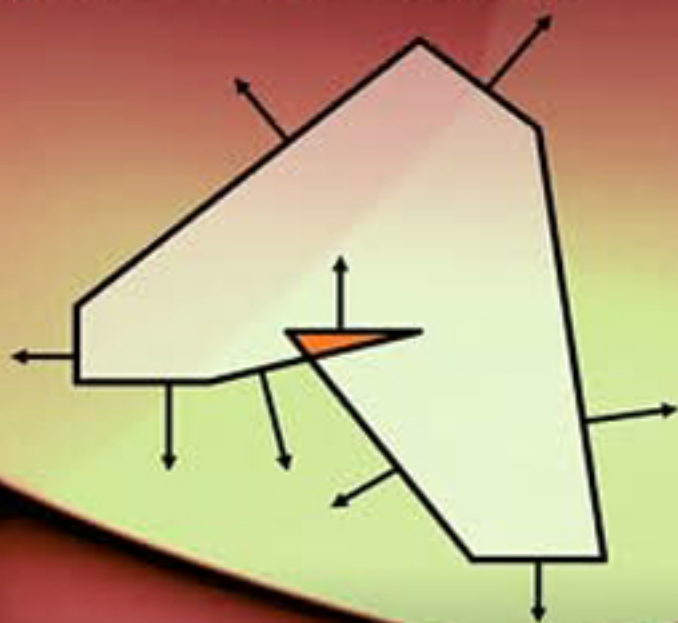


MATHEMATICS OF SHAPE DESCRIPTION

A Morphological Approach to
Image Processing and Computer Graphics



Pijush K. Ghosh | Koichiro Deguchi

 WILEY

MATHEMATICS OF SHAPE DESCRIPTION

A Morphological Approach to Image Processing and Computer Graphics

Pijush K. Ghosh

*National Centre for Software Technology
(now Centre for Development of Advanced Computing), India*

Koichiro Deguchi

*Graduate School of Information Sciences
Tohoku University, Japan*



John Wiley & Sons (Asia) Pte Ltd

MATHEMATICS OF SHAPE DESCRIPTION

A Morphological Approach to Image Processing and Computer Graphics

MATHEMATICS OF SHAPE DESCRIPTION

A Morphological Approach to Image Processing and Computer Graphics

Pijush K. Ghosh

National Centre for Software Technology

(now Centre for Development of Advanced Computing), India

Koichiro Deguchi

Graduate School of Information Sciences

Tohoku University, Japan



John Wiley & Sons (Asia) Pte Ltd

Copyright © 2008

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop, # 02-01,
Singapore 129809

Visit our Home Page on www.wiley.com

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as expressly permitted by law, without either the prior written permission of the Publisher, or authorization through payment of the appropriate photocopy fee to the Copyright Clearance Center. Requests for permission should be addressed to the Publisher, John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop, #02-01, Singapore 129809, tel: 65-64632400, fax: 65-64646912, email: enquiry@wiley.com.sg

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The Publisher is not associated with any product or vendor mentioned in this book. All trademarks referred to in the text of this publication are the property of their respective owners.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 42 McDougall Street, Milton, Queensland 4064, Australia

John Wiley & Sons Canada Ltd, 6045 Freemont Blvd, Mississauga, ONT, L5R 4J3, Canada

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data

Ghosh, Pijush K.

Mathematics of shape description / Pijush K. Ghosh, Koichiro Deguchi.
p. cm.

Includes bibliographical references and index.

ISBN 978-0-470-82307-1 (cloth)

1. Geometry, Algebraic. 2. Minkowski geometry. 3. Image processing—Mathematical models. I. Deguchi, Koichiro. II. Title.

QA565.G48 2008

516.3'5—dc22

2007051872

ISBN 978-0-470-82307-1 (HB)

Typeset in 10/12pt Times by Thomson Digital, Noida, India.

Printed and bound in Singapore by Markono Print Media Pte Ltd, Singapore.

This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

*To
Gopa, Nairita,
and Kazuko*

To the memory of Pijush K. Ghosh

The doors to knowledge were opened to me at an early age by my father. He taught me that the thirst for knowledge is unquenchable. He was like my very own magician, who made learning creative and fun. Today he is no longer by my side to guide me, but the fulfillment of his dream in the form of this book brings me immense joy. He is alive to me in the pages of this book, and this book is a ray of light in the darkness he has left behind in my life.

Daddy's Little Girl ...
Nairita Ghosh

Contents

Foreword	xiii
Preface	xv
1 In Search of a Framework for Shape Description	1
1.1 Shape Description: What It Means to Us	1
1.2 Pure versus Pragmatic Approaches	3
1.3 The Influence of the Digital Computer on Our Approach to Shape Description	5
1.4 A Metamodel for Shape Description	6
1.4.1 A Mathematical Model for Shape Description and Associated Problems	6
1.4.2 The Need for a Metamodel	8
1.4.3 Reformulating the Metamodel to Adapt to the Pragmatic Approach	12
1.5 The Metamodel within the Framework of Formal Language	16
1.5.1 An Introduction to Formal Languages and Grammars	17
1.5.2 A Grammar for the Constructive Part of the Metamodel	20
1.5.3 An Exploration of Shape Description Schemes in Terms of Formal Language Theory	20
1.6 The Art of Model Making	25
1.6.1 What is the Meaning of “Model”?	25
1.6.2 A Few Guiding Principles	25
1.7 Shape Description Schematics and the Tools of Mathematics	37
1.7.1 Underlying Assumptions when Mapping from the Real World to a Mathematical System	37
1.7.2 Fundamental Mathematical Structures and Their Various Compositions	39
2 Sets and Functions for Shape Description	43
2.1 Basic Concepts of Sets	43
2.1.1 Definition of Sets	43
2.1.2 Membership	44
2.1.3 Specifications for a Set to Describe Shapes	44
2.1.4 Special Sets	45
2.2 Equality and Inclusion of Sets	45

2.3	Some Operations on Sets	47
2.3.1	The Power Set	47
2.3.2	Set Union	48
2.3.3	Set Intersection	48
2.3.4	Set Difference	48
2.3.5	Set Complement	48
2.3.6	Symmetric Difference	49
2.3.7	Venn Diagrams	49
2.3.8	Cartesian Products	50
2.4	Relations in Sets	52
2.4.1	Fundamental Concepts	52
2.4.2	The Properties of Binary Relations in a Set	53
2.4.3	Equivalence Relations and Partitions	55
2.4.4	Order Relations	57
2.5	Functions, Mappings, and Operations	59
2.5.1	Fundamental Concepts	59
2.5.2	The Graphical Representations of a Function	62
2.5.3	The Range of a Function, and Various Categories of Function	65
2.5.4	Composition of Functions	66
2.5.5	The Inverse Function	68
2.5.6	The One-to-One Onto Function and Set Isomorphism	71
2.5.7	Equivalence Relations and Functions	72
2.5.8	Functions of Many Variables, n -ary Operations	74
2.5.9	A Special Type of Function: The Analytic Function	75
3	Algebraic Structures for Shape Description	77
3.1	What is an Algebraic Structure?	77
3.1.1	Algebraic Systems with Internal Composition Laws	79
3.1.2	Algebraic Systems with External Composition Laws	81
3.2	Properties of Algebraic Systems	83
3.2.1	Associativity	84
3.2.2	Commutativity	84
3.2.3	Distributivity	84
3.2.4	The Existence of the Identity/Unit Element	85
3.2.5	The Existence of an Inverse Element	86
3.3	Morphisms of Algebraic Systems	87
3.4	Semigroups and Monoids: Two Simple Algebraic Systems	92
3.5	Groups	94
3.5.1	Fundamentals	94
3.5.2	The Advantages of Identifying a System as a Group	100
3.5.3	Transformation Groups	101
3.6	Symmetry Groups	103
3.6.1	The Action of a Group on a Set	103
3.6.2	Translations and the Euclidean Group	105
3.6.3	The Matrix Group	106

3.7	Proper Rotations of Regular Solids	107
3.7.1	The Symmetry Groups of the Regular Solids	107
3.7.2	Finite Rotation Groups in Three Dimensions	112
3.8	Rings	112
3.8.1	Definitions and Examples	113
3.8.2	Some Classes of Rings	116
3.8.3	The Ring of Quaternions and Rotation of Objects	118
4	Morphological Models for Shape Description and Minkowski Operators	125
4.1	The Objective of Shape Description Modeling	125
4.2	The Basic Idea of Model Description	127
4.2.1	The Model	127
4.2.2	The Shape Operator	128
4.3	The Mathematical Nature of the Shape Operators	132
4.3.1	The Minkowski Addition Operator	133
4.3.2	The Minkowski Decomposition Operator	135
4.4	A Few Reasons for Choosing Minkowski Operators as Shape Operators	139
4.4.1	A Natural Description Tool	139
4.4.2	The Large Domain of the Model	140
4.4.3	Conciseness in Shape Representation	143
4.4.4	The Geometric Nature of the Shape Operators	144
4.5	Geometric Modeling by Minkowski Operations	145
4.5.1	Better Shape Representation	145
4.5.2	A Procedural Model	146
4.5.3	The Internal Structure of a Model	147
4.5.4	Concise Representation	148
4.6	Image Analysis by Minkowski Operations	150
4.6.1	Mathematical Morphology	150
4.6.2	Morphological Operators	151
4.6.3	Morphology of Multivalued Figures	154
4.6.4	Morphological Expansion	155
4.6.5	The Morphological Skeleton and its Properties	156
4.6.6	Morphological Decomposition of Figures	158
4.7	The Wealth and Potential of the Minkowski Operators	163
4.7.1	Minkowski Operations on Discrete Shapes	163
4.7.2	Minkowski Operations on Dynamically Varying Shapes	163
4.7.3	Inverse Shapes	164
5	Arithmetics of Geometric Shape	165
5.1	The Motivation for a Shape Arithmetic	165
5.1.1	Does Negative Shape Exist?	165
5.1.2	What Form Must Negative Shapes Take?	166

5.2 Morphology and the Theory of Numbers	167
5.2.1 Morphology for High-Level Vision	167
5.2.2 The Resemblance between Morphology and the Theory of Numbers	168
5.3 Boundary Representation by Support Functions for Morphological Operations	169
5.3.1 The Support Function Representation	169
5.3.2 The Support Function is a Signed Distance	170
5.3.3 From Support Function Representation to Boundary Representation and Vice Versa	172
5.3.4 Necessary and Sufficient Conditions for a Function to be a Support Function	173
5.4 Geometric Operations by Means of Support Functions	174
5.4.1 MAX and MIN Operations (Convex Hull and Intersection)	174
5.4.2 Morphological Operations in Boundary Representation	177
5.5 Morphological Operations on Convex Polygons	178
5.5.1 Computation by Means of Support Function Vectors	178
5.5.2 Computation by Means of Edges: The Emergence of the Boundary Addition Operation \oplus	181
5.5.3 Computation by Means of Slope Diagrams: The Unification of Minkowski Addition and Decomposition	182
5.5.4 The Computation of Boundary Addition	183
5.6 In the Domain of Convex Polyhedra	186
5.6.1 Computation by Means of Faces	186
5.6.2 The Slope Diagram Representation of a Convex Polyhedron	188
5.6.3 Computation by Means of Slope Diagrams	192
6 Morphological Operations on Nonconvex Objects	195
6.1 Problems with Nonconvex Objects	195
6.1.1 A Localized Definition of $F(A, u)$	195
6.1.2 The Anomalous Behavior of the Outer Normals at the Nonconvex Faces	196
6.1.3 The Need to Maintain Explicit Topological Information about the Operands	197
6.2 Slope Diagrams for Nonconvex Polygons	198
6.2.1 The Boundary Addition of Nonconvex Polygons by Means of Slope Diagrams	198
6.2.2 Boundary Operations on Nonconvex Polygons – More Complex Cases	201
6.2.3 Nonconvex Polyhedra and the Slope Diagrammatic Approach	205
6.3 A Unified Algorithm for Minkowski Operations	205
6.3.1 The Unified Algorithm	205
6.3.2 A Complexity Analysis of the Unified Algorithm	207
6.3.3 Simplification of the Unified Algorithm Depending on the Type of Input	208

7 The Morphological Decomposability and Indecomposability of Binary Shapes	215
7.1 The Morphological Indecomposability Problem	215
7.1.1 The Problem and its Motivation	215
7.1.2 Earlier Works	217
7.2 A Special Class of Binary Shapes: The Weakly Taxicab Convex (WTC) Polygons	219
7.2.1 Transforming Binary Images into Polygons	219
7.2.2 The Weakly Taxicab Convex Class of Polygons	220
7.2.3 A Few Properties of WTC Polygons Related to Minkowski Operations	223
7.3 Computing Minkowski Operations on WTC Polygons	226
7.3.1 Representation of WTC Polygons	226
7.3.2 The Minkowski Addition of Two WTC Polygons	229
7.3.3 The Minkowski Decomposition of Two WTC Polygons	234
7.4 A Few Results on Indecomposability in the WTC Domain	234
7.4.1 The Number of Indecomposable Shapes	234
7.4.2 Identifying Indecomposable Polygons within the WTC Domain	236
7.4.3 Simple Indecomposability Tests	241
7.5 A Brief Summing Up	242
7.5.1 Why Does the Uniqueness of Shape Decomposition Fail?	243
7.5.2 How Many Indecomposable Shapes are There?	244
7.5.3 How Can We Define New Equivalence Classes of Polygons?	244
7.5.4 Can We Devise Laws of Exponents, and Eventually Binomial Formulas for Shapes?	244
References	247
Index	251

Foreword

The computer description of shape and the computer manipulation of shape is complex simply because shape itself is complex. Of course, if the world of shape were limited to the Euclidean shapes, there would be no such complexity. However, shape includes all the varieties of biological shapes: from the shapes of trees and their leaves to fish, animals, flowers, and plants – and also natural shapes, such as those of coastlines, and of rocks and crystals.

Mathematical morphology is the mathematical study of shapes through a particular algebra of operations, known as the Minkowski set operations. Here, a shape can be thought of in the most general way possible, as a set of points in two or three dimensions. To fully understand the nature of the algebra of mathematical morphology requires: (1) an understanding of what an axiom system actually provides; (2) fluency in a variety of concepts associated with sets, including the set builder notation in mathematics; and (3) fluency in the concepts of algebraic structures. It is in this setting, formulated by Professor Deguchi, that the particulars of the concepts of mathematical morphology can most fully be appreciated.

Mathematics of Shape Description is the first book to devote half of its pages, in a tutorial fashion, to the basic background and/or essential preliminary concepts that lead up to the definitions of the mathematical morphological operators. This treatment of mathematical morphology simultaneously handles the discrete and the continuous domains, and is based on the mathematical morphology papers of Pijush Ghosh.

I knew Pijush Ghosh in the early 1990s, when he came to visit my laboratory at the University of Washington. His knowledge and understanding of mathematical morphology operations and what could be done with them, and what structures to use to implement continuous domain morphology in a computer program, was thorough and complete. I learned a great deal from him. He was a beautiful person, with a wonderful mind. He passed from this world prematurely, at an early age, only a few years after he returned to India, and he is greatly missed.

Robert M. Haralick
Distinguished Professor of Computer Science
Graduate Center, The City University of New York

Preface

In this book, the coauthors have set out to provide a shape description scheme that is a notational system for expressing the shapes of objects. This is also a way of writing the shape information symbolically to avoid both ambiguity and obscurity, just as we use notation to express music or electronic circuitry. We were interested in the algebraic structure hidden in the shapes, and we wanted to answer the following question: “Even if we identify that a given set of objects possesses an *algebraic structure*, how much is gained in practice from this discovery?” Of course, we have come to know that the set of objects is closed under some algebraic composition law, and that if it becomes possible to identify its set of generators, we may construct the whole set from that subset. However, can we conceive of some “stronger” kind of structure than this?

In this book, we take a morphological and set-theoretic approach to answering the above question. Then, we show the capability of this approach for image processing and computer graphics by presenting a simple shape model using two basic morphological and set-theoretic shape operators, which are called Minkowski addition and decomposition. The mathematical characteristics of these operators and their significance are explored in some detail, with the aim of eventually arriving at a formal theory of shape description.

We start the book with the mathematical basis of sets and functions, and next review modern algebra in general, thus highlighting the importance of the Minkowski operators. Then, on the basis of these preparations, we set out to construct a systematic method for the representation and analysis of shapes.

The first author, Pijush Ghosh, was a leading researcher in the area of Minkowski algebra and its applications to shape analysis and related problems.

His idea was to answer a simple question: “*Is it possible to do addition and subtraction (or multiplication and division) with geometric shapes as we do in ordinary arithmetic with numbers?*” In other words, given a geometric shape, does its inverse – that is, its *negative shape* – exist? If this were possible, then we might have obtained a remarkable insight into analyzing and synthesizing shapes, just as we have in the case of numbers.

This notion still fascinates me. We discussed the central problem in the understanding of shape, which can be compared with the analogous problem in number theory: “*Given a positive integer number n , are there integers k and $l > 1$ such that $n = k \times l$?*” As is well known, this question gave rise to one of the most fundamental concepts of number theory; namely, the concept of prime numbers.

Analogously, there exist sets of points in the plane or space that cannot be expressed as a Minkowski sum in any manner other than the most trivial one; that is, as the sum of a

single point and the given set S itself – in other words, they cannot be decomposed further, as a Minkowski sum of two simpler shapes. Such point sets may be termed morphologically indecomposable shapes, or *prime shapes*.

Then, one may ask, what shape can be considered to be the “*prime shape*”?

Before he was able to solve this problem completely, Pijush K. Ghosh passed away in 1999, at the age of 47, due to a brain tumor.

This book is a collaborative work between a mathematician, Pijush Ghosh, and an engineering researcher, Koichiro Deguchi. We first met in the early 1990s, at Professor Haralick’s laboratory at the University of Washington, where we were both visiting researchers. In the course of our discussions, it became clear that we both considered that image processing and computer vision were vitally important fields of information science and technology. Researchers in several areas of mathematics have contributed to the essential progress of these fields but, unfortunately, the ties between engineering and mathematics are not sufficiently strong, even in countries where image processing technologies have made considerable progress.

We decided to write a textbook to introduce a proper and well-defined algebra for image processing problems, and we began work in 1997. Sadly, we lost Pijush Ghosh halfway through our coauthorship, and his grand plan and our framework for the book were left in my hands. It took several years for me to restart the process of compiling and shaping his ideas in order to complete our task.

The first half of this book is my realization of Pijush’s original idea, whereas in the latter half of the book I have reconstructed Pijush’s original research.

Pijush’s family, and many of his friends and colleagues, have given me great encouragement throughout. I thank Professor Robert M. Haralick for his Foreword to this book. Dr. S.P. Mudur, of the National Centre for Software Technology, India, very kindly provided me with Pijush’s remaining manuscripts. The book would have been incomplete without the help of Pijush’s former colleagues Dr. Vinod Kumar and Ms. Sandhya Desai; and his friends Professor Subir Kumar Ghosh, of the Tata Institute of Fundamental Research, India, and Professor Kokichi Sugihara, of the University of Tokyo, Japan, have also been very supportive. Students in my laboratory at Tohoku University have also helped by proof-reading the book. I am most grateful to all of them.

Koichiro Deguchi

1

In Search of a Framework for Shape Description

1.1 Shape Description: What It Means to Us

It is difficult to obtain a very precise meaning of *shape*. In the *Oxford English Dictionary*, the meaning of the word “shape” is given as follows:

Shape – external form or contour; that quality of a material object (or geometrical figure) which depends on constant relations of position and proportionate distance among all the points composing its outline or its external surface.

The dictionary meaning of the word “description” is as follows:

Description – setting forth in words; reciting the characteristics of; more or less complete definition.

The dictionary meaning of “shape” emphasizes the fact that we human beings are aware of shapes through *outlines* and *surfaces* of objects, both of which can be visually perceived. On the other hand, shape does not take into account the color or texture of a surface. In more technical terms, the shape of an object is: “Information about the geometrical aspects of the surface of the object.” Shape description, therefore, involves specifying the information through a scheme or a system. A shape description scheme is a *notational system* for expressing the shapes of objects, a way of writing the shape information symbolically to avoid both ambiguity and obscurity, just as we use notation to express music or electronic circuitry.

It is well known that the discipline of shape description covers a very wide area, ranging from geometry to physics, and also to many other branches of science. For example, consider the task of describing the shape of a flat-faced solid cube. (a) A simple and direct scheme is to describe the shape by means of its vertices, edges, and faces. A vertex can be represented by a point (x_i, y_i, z_i) in a coordinate space, an edge by its end-point vertices, and a face by

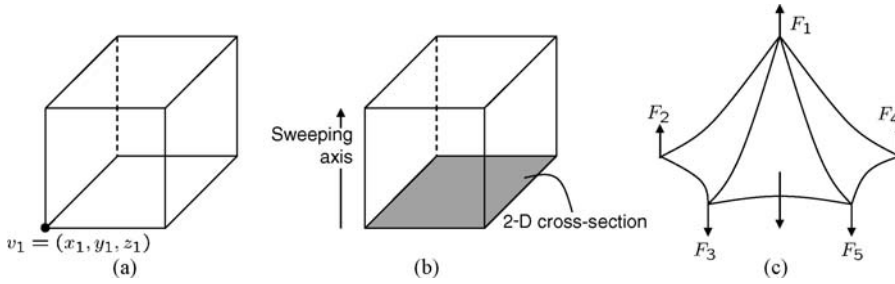


Figure 1.1 Shape can be described in a variety of ways

its bounding edges (Figure 1.1(a)). (b) The same cube can be described as the shape swept by a two-dimensional square cross-section when moved along a straight-line axis perpendicular to the cross-section (Figure 1.1(b)). This description is less direct than the previous one. (c) Sometimes, shape description may be more indirect. Consider the shape of a large suspension structure such as a tent. Its shape can be more conveniently described by means of the physical forces that act on various points of the tent (Figure 1.1(c)). The role of physical forces in describing the shapes of natural objects such as clouds, crystals, or trees is well known.

Such direct and indirect schemes are somewhat analogous to the *enumerative* and *generative* schemes in mathematics. A direct scheme is like writing down or enumerating all the elements of a set, such as

$$X = \{2, 4, 6, 8, 10, \dots\}. \quad (1.1)$$

An indirect scheme, on the other hand, is the specification of a set by defining a generating function for its elements, such as

$$X = \{x \mid x = 2y \text{ for } y \in \{\text{Natural numbers}\} \text{ and } y \neq 0\}. \quad (1.2)$$

It is impossible for us to cover the whole range of shape description. We have decided to restrict ourselves only to that part of shape description that is connected to geometry and other closely related concepts. Thus modes of description such as those depicted in Figures 1.1(a) and (b) fall within the scope of this book, but not modes such as shown in Figure 1.1(c).

We can be a little more precise in delineating the scope of the book. The shapes around us can broadly be divided into two categories: (1) shapes of manufactured objects or potentially manufactured objects (that is, well-designed objects); and (2) shapes of natural objects. The reason for this subdivision is that the mathematical techniques that are very well suited for description of the shapes of manufactured objects turn out to be inadequate for the shapes of natural objects. It is obvious that, “clouds are not spheres, mountains are not cones, coastlines are not circles, and bark is not smooth, nor does lightning travel in a straight line” [64]. Blum [7] wrote, “Euclid goes from triangles to more complex rectilinear objects, polygons. The only seriously considered nonpolygon is the circle. Where are the objects of biology? Where is the kidney bean, the tadpole? Note that the latter wiggles and is not congruent with or similar to even itself.” When attempts are made to describe them in terms of classical geometry and

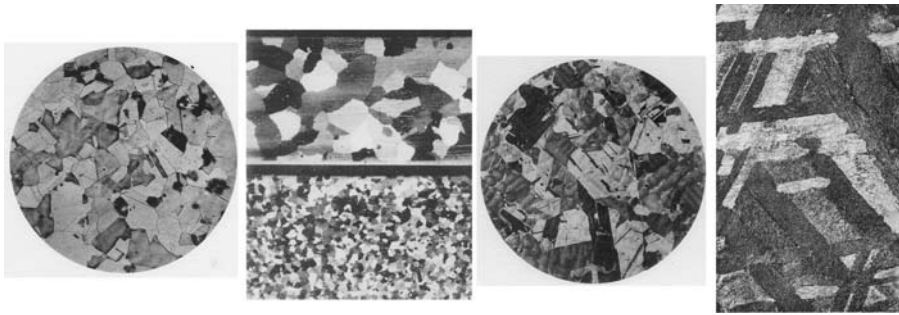


Figure 1.2 Shapes in crystals

mathematics, natural shapes turn out to be incredibly complex. Of course, there are some exceptions. For example, in nature, some crystals possess very regular geometric shapes, as shown in Figure 1.2.

But such exceptions are indeed very few in number. Thus if we restrict our domain to classical geometry and related mathematical areas, we are, in effect, addressing the question of shape description for manufactured objects. We, however, intend to discuss briefly the problems of shape description for natural objects, primarily to bring out the contrast.

1.2 Pure versus Pragmatic Approaches

Even after limiting ourselves to the geometrical aspects of shape, we find that the study of shape description starts from antiquity – from the geometry of Euclid or beyond, and extends to the geometric modeling, morphing, or fractal geometry of recent times.

The discipline of geometry itself has evolved from mankind’s pursuit of describing and measuring the shapes that are of immediate importance. The original motivation for geometry was to describe and measure land and buildings (the word “geometry” comes from the Greek *γημετρία*, which means “earth-measuring”). The famous Rhind papyrus, a copy of which has been preserved from the Hyksos epoch (about 1700 B.C.), testifies that at that time the Egyptians, although empirically, were able to calculate the area of a plane figure or volume of a solid. Even the early Greek mathematicians, such as Thales of Milet or Pythagoras of Samos (6th century B.C.), were more interested in practical problems of surveying (means of determining the boundary, size, position, etc.) and mensuration (means measuring). And, as it often happens, a field of study that started with a motivation to solve immediate and concrete practical problems transcended to a more abstract branch of science. It became far more rigorous, but moved further away from concrete ideas. The chief contributor to this transcendence is certainly Euclid. His work entitled *Elements* (*στοιχεῖα*), written in Alexandria in about 300 B.C., is still considered to be one of the most valuable scientific books of all time. And then there is a long list of outstanding mathematicians – Descartes, Gauss, Lobachevski, Bolyai, Riemann, Klein, and Hilbert, to mention a few – who have all contributed in an essential way to the development of this branch of mathematics called geometry.

However, in recent times we have seen a revival of the trend toward solving immediate practical problems concerning shapes of objects. Because of the advent of the digital computer, it

has become of interest to describe shapes so that they can be specified for the machine, manipulated, analyzed, reproduced as graphics output, and so on. Shape description is now a major field of study in the disciplines of computer vision, robotics, pattern analysis and recognition, computer graphics, image processing, and computer-aided design and manufacture. Subjects such as geometric modeling, computational geometry, and solid modeling are all concerned with shape description. But unfortunately, though naturally, in most of these modern developments the approaches to shape description are highly intuitive and pragmatic *ad hoc* responses to practical needs.

Thus, in the study of shape description we observe two distinctly contrasting trends. To illustrate the point, we could quote Hilbert [61]:

“On one hand, the tendency toward abstraction seeks to crystallize the logical relations inherent in the maze of material that is being studied, and to correlate the material in a systematic and orderly manner. On the other hand, the tendency toward intuitive understanding fosters a more immediate grasp of the objects one studies, a live rapport with them, so to speak, which stresses the concrete meaning of their relations.”

We can term these two trends the *pure approach* and the *pragmatic approach*.

1. *The pure approach.* The pure approach shows the following characteristics:

- The study of the subject itself is regarded as a self-sufficient exercise in pure thought. It is justified by its elegance and intrinsic beauty. In this approach, applications to practical problems are incidental and of lesser interest. “The mathematician does not study pure mathematics because it is useful; he studies it because he delights in it and he delights in it because it is beautiful,” remarked Poincaré.
- In this approach, the studies deal primarily with the question of “what,” rather than “how.”
- The studies are often very rigorous and logically well-connected with each other, but cannot be readily applied to solve practical problems.

2. *The pragmatic/utilitarian approach.* In contrast to the pure approach, a few of the characteristics of the pragmatic approach are as follows:

- A method or scheme is devised as a response to the needs of a particular problem. In this approach, application in the real world is of primary concern.
- The subject matter deals with the question of “how,” rather than “what.” The essential attitude is that “a number exists only if we know how to go about finding it.”
- The available methods are readily usable for the purpose of practical application, but are highly intuitive and, thereby, often lack rigor. Very often, the available methods appear like scattered techniques – disparate and, at first sight, unrelated.

Note: The difference between the pure and pragmatic approaches is much more profound and far-reaching than it may appear at first sight. The attitude of the truly pragmatic approach that “a mathematical object does not exist unless it can be constructed” leads to the rejection of many of the basic laws on which classical mathematics is based. This means that a large part of classical mathematics has to be abandoned, and a great deal of reshaping must be done. It is not possible to go into the detail of this discussion: see Bishop [6] and Heyting [40]. Here, we only present a simple example to clarify our point.

A truly pragmatic approach rejects the *law of excluded middle* in classical logic. This law asserts that every statement is either true or false. Therefore, according to classical logic, the statement S given below is always true:

S: Either there *exists* an even number n such that n is not expressible as the sum of two prime numbers, or, there *does not exist* an even number n such that n is not expressible as the sum of two prime numbers.

In the pragmatic approach, “there exists” is not sufficient; it is also necessary to “find/construct.” This means that if you want to follow the pragmatic approach, you also have to suggest a method to verify the statement S by demonstrating which of the two given alternatives is correct. The only method that can be suggested is to take every even number and to examine whether it is expressible as the sum of two prime numbers. Obviously, this method is not feasible, since the set of even numbers is an infinite set. Thus, according to the pragmatic approach, the truth or falsity of S cannot be verified and, therefore, S cannot be accepted as true, as is done in classical logic.

In this book, we shall attempt to look at the pragmatic approaches to shape description in the light of the pure approaches. To be little more precise at this point, our aim is to devise a broad mathematical system in which almost every recent scheme of shape description becomes a part. The mathematical system should provide a means of examining the shape description schemes more formally and establishing the logical thread that ties together these seemingly disparate techniques.

1.3 The Influence of the Digital Computer on Our Approach to Shape Description

Should the digital computer and computer programming influence our approach to shape description? On one hand they should, since one of our intentions – in fact, the primary intention – is to specify shapes of objects to the computer for the purpose of analysis, manipulation, rendering, and so on. On the other hand they should not, since any mathematical framework tends to describe things in more general and abstract terms, with a deemphasis on a specific “real” world, even though that real world might have provided the original motivation for these studies.

In the pragmatic approach, the dilemma is greater, because as soon as we say that “we can find/construct,” the question arises “By what can you construct?” The construction tools have a more profound influence on mathematical thought than it may seem. The decimal system of counting was invented because we have eight fingers and two thumbs on our hands, with which we can count. The ancient Greeks had a quite restricted notion of geometric constructibility in mind, since their construction tools were limited to the ruler and the compass. That is why the problem of trisecting an angle attracted so much attention, whereas the problem can be easily solved by the use of some very simple additional devices, such as a paper strip [5]. In recent times, the extensive use of spline curves, parametric cubic curves, and parametric surfaces in describing shapes is due to the fact that our present-day computer can handle them well.

This dilemma can be solved, although partially, if we remember that the *question of description often transcends several levels of detail*.

Consider, for example, the problem of describing the shape of a nonconvex polygon. The first-level decision may lead to describing it by means of nonoverlapping convex polygons. The decision at the second level may lead to describing each of the convex polygons by its vertices. At the third level the decision is to describe each vertex with the help of vectors. The fourth-level decision may lead to the choice of a pair of real numbers for each vector, in Cartesian coordinates. At the fifth level, which is based on the knowledge that data are to be stored in a computer, the decision may lead to a floating-point representation of each number, where a real number r is represented by a pair of integers denoting a fraction f and an exponent

e to a certain base, say 10 (such that $r = f * 10^e$). The sixth-level decision may lead to a binary representation, and so on. This example clearly demonstrates that the approaches in the initial levels are more abstract and are more influenced by the problem situation, whereas the later ones are progressively more dependent on the construction tool.

Therefore, in building up the mathematical framework for shape description, initially we start with the assumption that we are working on an *ideal computer* and we adopt an abstract mathematical approach to the problem. Only in the later stages do we consider various issues involving the implementation of various shape description schemes on a modern-day digital computer. The design of relevant computer algorithms, analysis of the complexities of the algorithms, consideration of their efficiencies, and so on are dealt with at that stage.

But what do we mean by an “ideal computer”? There are several equivalent ways of describing the notion of an ideal computer. The *universal Turing machine* concept, however, because of its machine orientation, appears to be a better model for an ideal computer than the other more abstract mathematical models. There several good texts on mathematical foundations of computer science in which the universal Turing machine and other related concepts are discussed [41]. It is not possible here, nor even urgently necessary, to get immersed in those details. For our purpose, it is sufficient to know at present the consequences of using the universal Turing machine, which are that any method suggested to find/construct something must have the following characteristics:

1. It must be *deterministic*; that is, we expect to obtain the same results from identical starting conditions.
2. It must be executable in *finite time* and it must use some *finite facility*. This condition implies that our resources for computation are finite.
3. The execution of such a method must be *mechanical*. This roughly means that the steps involved in executing the method are so precisely described that, in principle, even a mechanical device could execute the method.
4. The method can be cast in numerical terms. This is called the *arithmetization* of a method, and is, in fact, is a consequence of Property 3. We mention this property to emphasize the important fact that in digital computing we can always restrict, with no loss of generality, the objects of discussion to the natural numbers.

In computer terminology, such a method is known as an *effective procedure* or an *algorithm*.

Note: At present, by “construction tool” we mean only *physical* tools. But in the finer analysis we must also consider the *theoretical* tools that are readily available for construction. The discipline of analytic geometry has progressed very rapidly since the concept of coordinates transformed geometric problems into algebraic problems, and all the theoretical tools of algebra then became immediately available. We, however, do not delve further into such finer points.

1.4 A Metamodel for Shape Description

1.4.1 A Mathematical Model for Shape Description and Associated Problems

It is possible to express shapes of objects in various ways, through words, two-dimensional drawings, pictures, photographs, or some other means. But in all such modes of expression there lies a basic problem; there is always a finite probability of the presence of ambiguity, obscurity,

or impreciseness in the description. If a human being is asked to interpret these descriptions, he or she might fill the gaps and vagaries with common sense and general knowledge about objects. But then this does not become a *mechanical process*. We cannot expect a machine to do that – at least not at present.

A solution, therefore, is to express shapes of objects using a mathematical language. We know that any language basically consists of two parts – (i) a collection of entities called the *vocabulary* and (ii) a set of *rules*. In describing something that is not part of the language itself, we use the vocabulary of the language according to the rules. The difference between an ordinary language and a mathematical language is simply that while in the former the rules (the ordinary rules of grammar) and the vocabulary are not precisely defined, in the latter they are very precise. Therefore, by using a mathematical language to describe shapes, we attain preciseness and we can eliminate ambiguity and vagaries.

Even then, there are quite a few problems:

1. *Limited descriptive power*: There is no mathematical language that is as able to talk about such a wide variety of things outside itself as an ordinary language such as English. Try, for example, to describe in a mathematical language the shape of a square whose corners are rounded. You will find that even such a simple task becomes quite difficult. One remedy is to choose a mathematical language that is rich enough for the domain of shapes in which you are interested.
2. *Mapping problems from the real world to the abstract world*. There is another problem that is more subtle, but nevertheless important. What we express by a mathematical description is an abstract concept – an *idealized* description. The extent to which this idealized description can be usefully regarded as a description of the shape of an object that exists in the real world depends on the extent to which the characteristics of the real-world shape correspond to the characteristics of the idealized description. Let us give an example to clarify the point. Assume that we are studying the question of describing the area of a circle in terms of its radius. In a mathematical language, the correspondence between the area A and the radius r can be expressed as

$$A = \pi r^2. \quad (1.3)$$

Is the above equation a perfect description of the relationship between the area and the radius? It is not, because the above equation does not guarantee that the value of r will only be positive; according to the above equation, it may be positive as well as negative, but no circle with a negative radius exists in the real world. This means that the characteristics of this real-world problem correspond to the characteristics of that particular mathematical equation only in its positive quadrant, but not in the other quadrants.

3. *Specialized expressive characteristics*. Every language, including any mathematical language, possesses its own expressive characteristics. Take any language and you will find that it is able to express a few things more easily and naturally than others. Therefore, when we choose a particular mathematical language to describe shapes, the expressive characteristics of the chosen language will make certain shape information explicit at the expense of information that is pushed into the background and may be quite hard, although possible, to extract.

Consider, for example, the problem of representing a point on the plane. With respect to some chosen origin, the point, say p , can be represented as $re^{i\theta}$ in polar coordinates, or

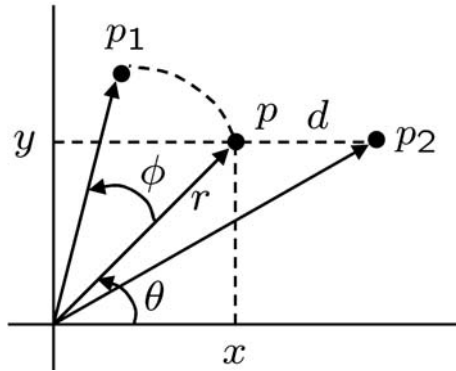


Figure 1.3 A point is represented in both the polar and the Cartesian coordinate systems; the advantages and disadvantages of both of the representation schemes are depicted schematically

as (x, y) in Cartesian coordinates (Figure 1.3). If p is rotated by an angle ϕ to the point p_1 , the representation in polar coordinates turns out to be more convenient; because in polar coordinates p_1 is expressed as $p_1 = re^{i(\theta+\phi)}$, while in Cartesian coordinates $p_1 = (x \cos \phi - y \sin \phi, x \sin \phi + y \cos \phi)$. On the other hand, if p is translated horizontally by d units to p_2 , the Cartesian system representation is more suitable since $p_2 = (x + d, y)$, while in polar coordinates $p_2 = ???$ (this is not simple – try to write down the form).

This issue is vitally important whenever we choose a particular mathematical language to describe shapes. In the pragmatic approach, every shape description scheme is *purpose-oriented*. Therefore, it is necessary to know “what aspects” of the shape information are important for the intended applications and “whether those aspects” will be explicit in the mathematical language that we have chosen.

Note: How information about an entity is represented can greatly affect how easy it is to do different things with it. The Arabic and the Roman numeral systems are both mathematical systems for representing numbers. Whereas it is easy to add, subtract, and even multiply in the Arabic system, it is not easy to do the same – especially multiplication – with Roman numerals. This is a key reason why the Roman culture failed to develop mathematics in the way the Arabic culture had.

To overcome these problems, we can approach the problem of shape description with the following objectives in mind:

- To study the characteristics of various mathematical systems that appear to be probable candidates for describing shapes.
- To examine carefully what the intended applications are and to extract the aspects of shape information that are needed for those applications.
- To choose the mathematical system, and the corresponding language, that appears to be the most appropriate for the applications in hand.

1.4.2 The Need for a Metamodel

The discussion in the previous subsection brings out something crucial. All shape description schemes are *purpose-oriented*. The choice of a description scheme must be guided by the problem context. No scheme is better than all others for all purposes.

Therefore, the mathematics of shape description first of all needs a common mathematical system, by means of which we can analyze various shape description schemes – the description domains of the schemes, conciseness in description, the precisions attainable, their effective areas of application, and so forth. We can loosely refer to such a mathematical system as a *metamodel* for shape description (meta – beyond, transcending, higher), since it will be a system with which to study the nature of various shape description models.

It can be shown that almost every existing shape description scheme follows a simple model. A scheme includes the concept of a set of *primitive shapes*, say P , and a set of conditions or *axioms*, say R . Every shape defined by the scheme is made out of those primitive shapes in such a way that the axioms are satisfied at all times. In other words, the structure of the set of shapes defined by a scheme is completely determined by P and R . In mathematical terms, a set endowed with some structure is usually called a *system*, or sometimes a *space*. In that sense, every description scheme defines a space S that is determined by P and R . We can denote this as $S = \langle P, R \rangle$.

Let us clarify the concept further by means of a few examples, some of which are constructed following Batten [4].

Example 1.1. The set P of primitive shapes includes *points* and *lines*. (We use the terms “point” and “line” in their commonly used geometric sense – that is, in the sense of Euclidean geometry – and therefore do not define them again.) The set R contains the following three axioms:

- 1. Every object of the space contains precisely six points and four lines.
- 2. Each line has two points.
- 3. Each point is on at most four lines.

It is not difficult to list the objects that are found in this space, which are shown in Figure 1.4. □

If you work out the objects for yourself by following the axioms, you will notice that the objects that satisfy axioms 1 and 2 also satisfy axiom 3. In fact, it is easy to see that axiom 3 follows immediately from axiom 1: if there are only four lines, then clearly no point can be on more than four lines. Such a set of axioms is said to be *dependent*, since one or more

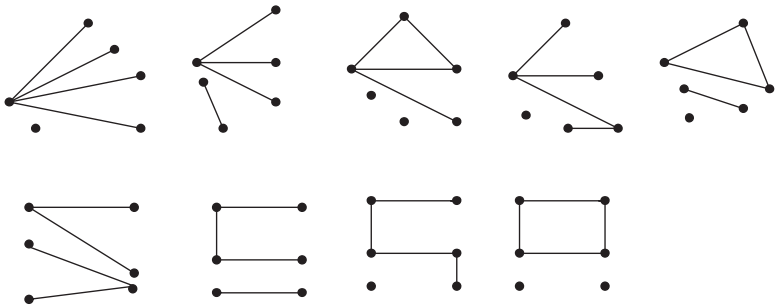


Figure 1.4 The objects in the space $S = \langle P, R \rangle$ as defined in Example 1.1

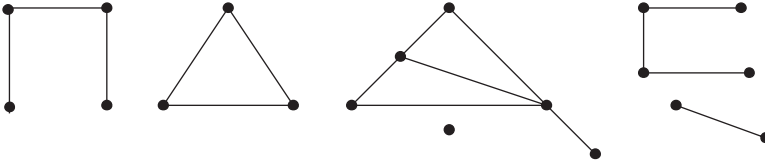


Figure 1.5 A set of figures are given whose space S has to be determined for Example 1.2

axioms of the system can be proved using the remaining axioms. Otherwise, the set of axioms is *independent*.

Example 1.2. Consider the figures presented in Figure 1.5.

Can we discover a set P and a set R that will define them all? It is not difficult to see that the set P of primitive shapes must include “points” and “lines.” Note that the task of devising a suitable set of axioms R is not very easy. It appears that the following set of axioms R are satisfied by all the objects:

1. Any line has at least two points.
2. Two points are on at most one line.

Obviously, in this case the space $\langle P, R \rangle$ contains many more objects than the given ones; and so in that sense, our proposed description scheme is not very concise. \square

Example 1.3. We again assume that the set P includes “points” and “lines” and that the set R includes the following axioms:

1. Every object of the space contains precisely six points and seven lines.
2. Each point is on one line.
3. Each line contains one point.

Try to construct the objects of the above space. You will find that it is not possible to construct any object that satisfies all three axioms. This is because we are trying to set up a one-to-one correspondence (by axioms 2 and 3) between sets with six points and seven points (from axiom 1), respectively. That is, of course, impossible. Such a set of axioms is called *inconsistent*. \square

Example 1.4. We also observe the same model in Euclidean geometry. The set P of primitive shapes includes only “point,” and a point has no constituent parts. The set R contains the following axioms:

1. A straight-line segment can be drawn joining any two points.
2. Any straight-line segment can be extended indefinitely in a straight line.
3. Given any straight-line segment, a circle can be drawn that has the segment as radius and one end-point as center.
4. All right angles are congruent.

5. If a straight line falling onto two straight lines makes interior angles on the same side that are less than two right angles, the two straight lines, if produced indefinitely, will meet on that side on which the angles are less than two right angles.

(This is a slightly simplified version and is not given in exactly the form that was set out by Euclid.)

Note: It can be added here for historical interest that many mathematicians were unhappy about Euclid's axiom 5 (also known as Euclid's "parallel postulate"), which certainly does not share the same grace and elegance as the other four. They felt that Euclid's set of axioms was not independent, and for a period of over 2,000 years countless attempts by untold numbers of people were made to prove that the parallel postulate could be derived from the others. It is not possible to present that exciting history here. But it is necessary to mention that it was only as recently as the 19th century that the parallel postulate was shown to be independent of Euclid's remaining axioms. The *Bolyai–Labachevskian geometry* (also called *hyperbolic geometry*) was invented precisely on the basis of the negation of the parallel postulate. It was in the year 1899 that David Hilbert, in the work *Grundlagen der Geometrie*, gave a precise system of the primitive notions and axioms of Euclidean geometry and a full proof of the consistency of the axioms. Any serious reader who wants to go through those interesting accounts can consult Borsuk and Szmielew [9]. \square

Let us summarize our discussions in this subsection:

- (a) Almost every shape description scheme is a system or space S that is defined by a set of primitive shapes P and a set of axioms R . We may, therefore, use the terms "description scheme" and "description system" as synonyms.
- (b) There are two points of view in devising a space S :
 - (i) Given a set P and a set R , find the space $S = \langle P, R \rangle$. To find the space means to find objects of the space, various properties of those objects, and so forth. Examples 1.1 and 1.4 are of this type. This is, in a sense, a pure approach to shape description.
 - (ii) Given a familiar space S (for example, some objects in the real three-dimensional space), find P and R such that $\langle P, R \rangle = S$. Example 1.2 is of this type. We adopt this viewpoint in the pragmatic approach to shape description.
- (c) The axioms of a system are like the rules of a game. Therefore, "Are the axioms true?" is a nonsensical question. If you change a set of axioms, you are no longer playing the same game. In Example 1.4 we mentioned that we obtained two different geometries – Euclidean geometry and hyperbolic geometry – just by changing the parallel postulate. But both of the geometries are equally correct from the standpoint of logic.
- (d) However, we can dispute the applicability of a set of axioms in any particular case. When we try to apply the theory to the real world, the question of whether or not the shapes in the real world work in the way the axioms predict is certainly pertinent. But it is not a question that is part of the theory. It needs to be answered by experimenting with those real-world shapes. (Thus the question of whether Euclidean or hyperbolic geometry better describes real space can be settled, if at all, only by way of experiment.)
- (e) In formulating a set of axioms, we must take care to ensure that the set of axioms is *consistent*, *independent*, and *complete*.

A set of axioms that does not contradict itself is said to be *consistent*. Through Example 1.3, we have demonstrated that unless the set R is consistent, we cannot construct any object of the space. As an aesthetic matter, it is also considered desirable that the axioms

should be *independent*. If one axiom of the system can be derived using the remaining axioms, there is not much point in including it in R , as is shown in Example 1.1. But we do not run into any great trouble even if we use a dependent set of axioms. *Completeness* is the complement to the notion of consistency. If a set of axioms is consistent, then it is possible to construct *at least one object* of the space that satisfies all of the axioms. Completeness works the other way round. A set of axioms is said to be complete if it is possible to construct *all the objects* of the space that we are attempting to describe. In Example 1.2, the set R is complete since we can describe all the objects that are given to us.

1.4.3 Reformulating the Metamodel to Adapt to the Pragmatic Approach

In the pragmatic approach, most often the task is to devise a suitable set of P and R such that a given space S can be “conveniently” described by them. The space S often contains shapes of objects that exist in the real world or that potentially exist in the real world (such as the design of a man-made object intended for production).

Consider some frequently encountered P and R . The set P may contain shapes such as points, lines, rectangular boxes, spheres, cylinders, and so on. The set R is a set of rules/conditions – which can take any form – that the objects of the space must satisfy. The concept of the axiom is very general – and precisely because of this utmost generality, we face problems. If our intention had been only to examine the properties of the objects of the space, as is often the intention in the pure approach, this could have been done by proving different theorems with the help of the axioms in R . But in the pragmatic approach not only we are interested about the object properties, but we also want to construct the objects of the space. Naturally, such a construction task becomes particularly difficult if the axioms are all *nonconstructive* in nature. In that case, as is pointed out in Section 1.2, the only method available is to construct all the possible objects with the help of the set P and then examine which of them satisfy the axioms in R . Obviously, this is not a feasible approach.

One way to partially reduce the problem is to choose the set of axioms R in such a way that at least a subset, say C , of R will be constructive in nature. In other words, R should be subdivided into two subsets, C and A , where

C : is the set of axioms that are constructive in nature; they can be termed as *construction rules* or *production rules*.

A : is the set of nonconstructive axioms in R ; they can be called *pure axioms* or *explicit axioms*.

The construction task now becomes simpler. Using the set of production rules C , we construct objects and examine which of these objects satisfy the axioms in A . We only have to examine those objects that can be constructed by means of P and C . Depending on C , such objects will be limited in number. It may even be possible at times to choose the set C in such a way that no other axiom is necessary. In such a situation, all the objects that could be constructed by means of C will be the objects of the space.

It must be pointed out that we are not introducing any new concept here. Many important sets in mathematics that have structures are defined in this constructive way. That is, in defining a set we are given certain primitive objects in the set and, in addition, we are given a few rules for forming new objects from old ones, or from objects already known to be in the set. The

defined set then consists of all objects that can be formed by starting with the primitive objects and repeatedly applying the rules for forming new objects. Such a method of defining a set, or a collection of objects, was called *genetic* by Hilbert [5]. Below, we give a couple of examples of the genetic method.

Example 1.5. The set $N = \{0, 1, 2, 3, \dots\}$ of natural numbers is unquestionably the most important mathematical system. There are many set of axioms defined by various mathematicians to describe the natural number system (you can consult any standard textbook on modern algebra). The simplest set of axioms for the natural number system, and one of the first, was published in 1889 by the Italian mathematician and logician Guiseppe Peano. The following is a version of Peano's axioms:

1. $0 \in N$.
(This is the set P of primitive elements.)
2. If $n \in N$, then $S(n) \in N$, where $S(n) = n + 1$.
($S(n)$ is generally known as the *successor function* and it maps a number n to $n + 1$.)
3. If a subset $U \subseteq N$ possesses the properties
 - (a) $0 \in U$, and
 - (b) if $n \in U$, then $S(n) \in U$
 then $U = N$.
(This is known as the *minimality property* and it asserts the fact that a minimal set satisfying (a) and (b) is the set of natural numbers.)

The above set of axioms is an excellent demonstration of the genetic method. All natural numbers N can be obtained by starting with the number 0 and repeatedly applying a rule that furnishes the successor of a number when we are given that number. \square

Example 1.6. The following example is constructed following Beckman [5]. He remarked that if we take *Genesis* literally, then the class of all human beings (in the past, present, and future) can be defined sharply by the following expression:

$$HUMAN ::= ADAM \mid EVE \mid CHILD(HUMAN, HUMAN). \quad (1.4)$$

The symbol " \mid " is to be interpreted as "or." In words, the above expression means that *ADAM* and *EVE* were the only two human beings at the beginning. According to our notation, $P = \{ADAM, EVE\}$. The set C contains only one production rule, which states that a *CHILD* of any two human beings is a human being. Note that this particular definition of *HUMAN* is *recursive*. The definition of an object is called recursive if it is defined in terms of itself. The term *HUMAN* appears on both sides of the defining expression. In this example, there is no explicit axiom A . \square

Note: We deviate from the main theme for a short while at this point, in order to decide on terminology. We observed that any production rule in C takes one or more old objects from the description space to produce a new object in the space. In mathematics, this is generally known as the *mapping* of old objects to new ones. Mappings are also called *functions*. This is why $S(n)$ is called the successor function and $CHILD(HUMAN, HUMAN)$ is also expressed in the conventional functional form. We can also take the view that some operation is being performed on old objects to produce a new object. In that sense, the functions can also be called *operators*. The addition of two numbers x and

y , for example, can either be represented with the help of the addition operator “+” or with the help of a function “ADD”; that is, either as $x + y$ or as $ADD(x, y)$. Thus the terms *mapping*, *function*, and *operator* could be considered as synonyms for our present purpose.

Note carefully that we use the term “operator” in its widest mathematical sense. For example, we consider “ $\max(a, b)$ ” as an operator that operates on two real numbers, a and b , to produce the larger of the two. We can go further and define more powerful operators using the operators defined earlier, and so on. (For example, consider the definition of the operator called “*largest*”: $largest(a, b, c) = \max(\max(a, b), c)$.)

In what follows, we prefer to use the term “function/operator” rather than “mapping.”

Using this new terminology, we can summarize the discussion of this subsection. It is shown that in the pragmatic approach to shape description, a metamodel should consist of the following three units:

1. A set of *primitive shapes*, say P , and a set of functions/operators, say $*$. These operators can be called *shape operators*.
2. A set of *production rules*, say C , which specify how the shape operators are to be used to construct new shapes from the already existing shapes.
3. A set of *explicit axioms*, say A , which specify conditions that each constructed shape must satisfy. In a sense, A is a set of *constraints* or *restrictions*. In a shape description scheme, the set A may or may not be present.

For easy reference, we will denote our metamodel by the notation $(P, *, C, A)$.

We will now demonstrate below, by means of a simple example, how a shape description scheme can be built from the above notion of a metamodel.

Let us call this scheme the *Toy system*, and let it include the following primitive shapes:

$$P = \{\text{an unbounded plane, all the directed straight lines } \vec{l}_i \text{'s on the plane}\}. \quad (1.5)$$

At present, by “unbounded” shape we mean a shape that is not definable within a finite space. A more rigorous concept of boundedness will be given later.

We know that any straight line divides a plane into two unbounded regions; each region is called a *half-plane* (Figure 1.6(a)). Now there must be some mechanism to distinguish between the two half-planes. There are several ways to do this, but the simplest way is to give a orientation/direction to the straight line, so that one of the half-planes is said to lie on its left while the other one lies on its right. We can call them the left half-plane and the right half-plane, respectively.

In the *Toy system*, the set $*$ of shape operators contains two operators:

$$* = \{\text{half-plane}_{\text{left}}, \text{intersection}\}. \quad (1.6)$$

Let the set C include two production rules:

$$\text{region}_i ::= \text{half-plane}_{\text{left}}(\vec{l}_i), \quad (1.7)$$

$$\text{intended_shape} ::= \text{intersection}(\text{region}_1, \text{region}_2, \dots, \text{region}_n). \quad (1.8)$$

Here, $\text{half-plane}_{\text{left}}(\vec{l}_i)$ is a unary operator that generates the left half-plane specified by the straight line \vec{l}_i and *intersection* is an n -ary operator, where n is a finite integer.

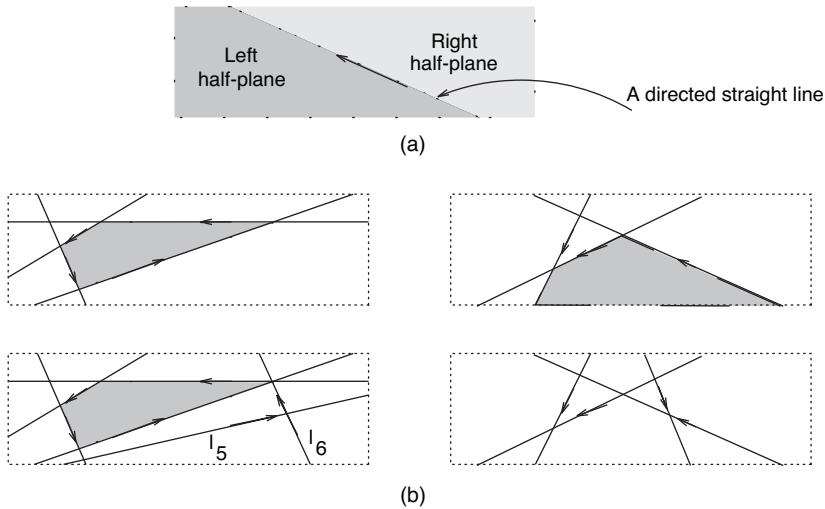


Figure 1.6 (a) A straight line l_i divides the plane into two half-planes – for a directed straight line, we can distinguish the left and right half-planes. (b) A few of the shapes that are described by our *Toy system* – the second shape shown in the figure is unbounded

Let us initially assume that there is no explicit axiom A in the *Toy system*.
In Figure 1.6(b), we show some of the shapes that belong to the *Toy system*.
What are the characteristics of this *Toy system*? In other words, what are the answers to questions of the following kind:

1. *Description domain*. For what class of shapes is the system designed?
 2. *Uniqueness in description*. In this system, is there more than one way to describe the same shape?
 3. *Geometric and topological properties*. What are the (simple) geometric and topological properties of the objects that are being constructed?
 4. *The physical validity of the shapes*. Is every shape constructed by the system physically valid?
 5. *Areas of application*. What are the most suitable areas of application for the system?
- With regard to the description domain, take, for example, some given shapes shown in Figure 1.7. Which of them can be described by our *Toy system*? It is obvious that the system

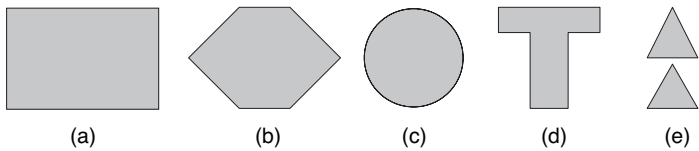


Figure 1.7 A given set of shapes. Which of them can be described by our *Toy system*?

can describe only polygonal regions. This means that the circular shape (Figure 1.7(c)) cannot be represented. But it may not be so obvious that only “convex” polygonal regions can be described by our system. This follows from two facts: (1) any half-plane is convex set; and (2) the intersection of two convex sets is also convex. This means that out of the five given shapes, only the first two belong to the *Toy system*.

- Descriptive uniqueness is a desirable criterion for assessing the equality of objects. If a given shape can be described in several ways in a system, the description is called nonunique. Consider the two shapes shown on the left-hand side of Figure 1.6(b). Both of the polygons are exactly the same, though in the second case two more half-planes are used (by means of lines \vec{l}_5 and \vec{l}_6), which are redundant. This means that the *Toy system* does not guarantee uniqueness in description. To achieve descriptive uniqueness, it is necessary to introduce some additional axioms (perhaps, by means of a set A) in the system to eliminate every redundant half-plane.
- To make use of the *Toy system* for any practical purpose, it is essential to know some of the simple geometric and topological properties of the objects that belong to the system. For example, does our system allow the description of a shape that consists of two or more disconnected convex components, as shown in Figure 1.6(e)? An easy topological analysis shows that our system generates only *simply connected* shapes.
- The shape shown in the upper right-hand corner of Figure 1.5(b) is a convex polygonal region, but it is unbounded. The fourth shape shown in the same example is an empty set. We can, therefore, conclude from these examples that the *Toy system* does not ensure that every shape constructed by the system is physically valid. However, the notion of physical validity of a shape is rather intuitive and to a large extent application dependent. To specify the concept of “valid shapes” required for some particular application, we feel the need of a set of explicit axioms. Depending on the requirements of the intended application, the notion of physical validity of a shape could be reformulated as a set of mathematical conditions that a valid shape must satisfy, and this set of conditions has to be included in the system as the set A .
- It is difficult to answer, in a clear-cut way, what areas of application are most suitable for the *Toy system*. It is desirable to view an area of application as a set of operations to be performed on the given set of objects. Since any shape in this system is described in terms of intersections of half-planes, the natural areas of application are mostly those where the intersection of objects is involved. The hidden line removal problem in computer graphics, or the problem of cutting a polygonal object out of a sheet of material, appear to be problems of this type.

1.5 The Metamodel within the Framework of Formal Language

Instead of viewing the metamodel as an *axiomatic system*, it is also possible to view it as a *formal language*. The formal language viewpoint is advantageous for various reasons, which will be apparent from our subsequent discussion.

We assume here that you are familiar with the concept of formal languages and grammars. However, for the sake of completeness we will give a brief introduction to the notion of formal languages.

1.5.1 An Introduction to Formal Languages and Grammars

The original motivation for formal languages was the description of natural languages. A sentence in a language has a structure. Consider the sentence “a student reads the book.” Its structure is shown in Figure 1.8(a). The diagram displays the syntax of the sentence in a manner similar to a tree, and is, therefore, called a *syntax tree*. On the other hand, we can also consider a similar structure as shown in Figure 1.8(b). Our justification for introducing the formal languages and grammars into image analysis is just the similarity of this structure.

This structure can also be specified by using the symbols

$S :< sentence > \quad V :< verb > \quad O :< object > \quad A :< article >$
 $N :< noun > \quad SP :< subject phrase > \quad VP :< verb phrase >$
 $NP :< noun phrase >$
 $\circ : \text{concatenation operator; that is, juxtaposition of adjacent element}$

in the following rules:

$S \longrightarrow SP \circ VP,$
 $SP \longrightarrow A \circ N, \quad VP \longrightarrow V \circ O,$

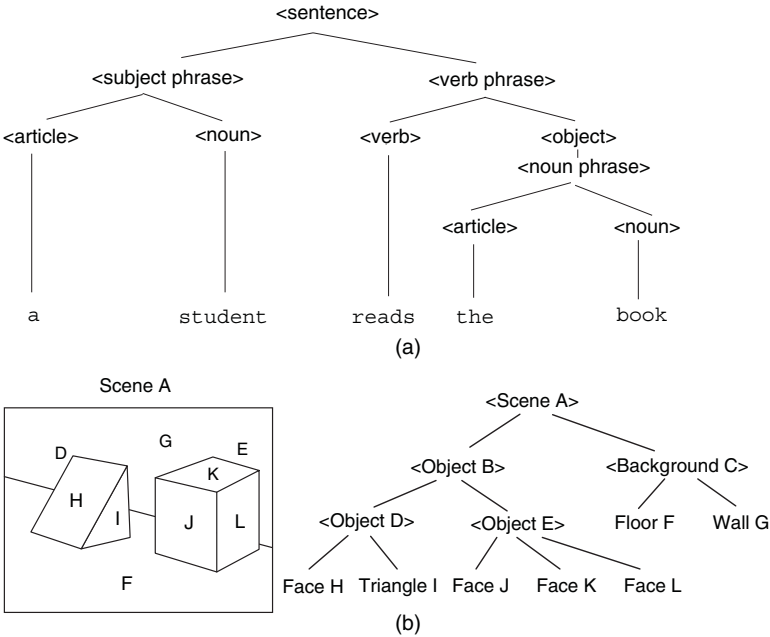


Figure 1.8 The structure of (a) the sentence “a student reads the book” and (b) the scene, by means of a syntax tree

$$\begin{aligned}
O &\longrightarrow NP, \\
NP &\longrightarrow A \circ N, \\
A &\longrightarrow a|the \quad N \longrightarrow \text{student}|\text{book} \quad V \longrightarrow \text{reads}.
\end{aligned}$$

The above rules state that a “sentence” is composed of a “subject phrase” followed by a “verb phrase”; the “subject phrase” is composed of an “article” followed by a “noun”; a “verb phrase” is composed of a “verb” followed by an “object”; and so on. Finally, it is stated that an “article” may be either *a* or *the*; a “noun” may be either *student* or *book*; and so on.

A few general characteristics of this method of specifying the syntax of a sentence can be noted at this point. There are some symbols, such as *a*, *the*, *student*, *book*, and *reads*, that correspond to words from the English dictionary. These symbols can be called *terminal symbols*, since they appear at the terminal of a syntax tree. The terminal symbols are indivisible tokens in a language. There are also some linguistic concepts, such as subject phrase, verb phrase, verb, and so on, which can be called *nonterminal symbols*. The nonterminal symbols are the intermediate steps in describing the syntax structure of a sentence. There is a special nonterminal symbol, called a *sentence*, that is the *starting symbol*. Moreover, there is a set of *production rules* that state how the terminal and non-terminal symbols have to be concatenated together to form a sentence. This method of specifying the syntactic definition of the language is generally called the *grammar* of the language.

We can formalize the idea of a grammar in the following way.

Definition 1.1: A (*phrase-structured*) grammar is defined by a 4-tuple $G = (V_T, V_N, S, \Phi)$, where V_T and V_N are sets of terminal and nonterminal symbols, respectively. S , a distinguished element of V_N , is called the starting symbol or sentence. Φ is a set of production rules, which have the general form $\alpha \longrightarrow \beta$, where α, β are strings over $V_T \cup V_N$, and “ \longrightarrow ” is read as “is replaced by.” The only operation between the strings is concatenation \circ which, for most of the time, is not explicitly stated. All syntactically correct sentences generated by applying the rules of G are called the *language* $L(G)$ of G . The set $V_T \cup V_N$ is called the *vocabulary*, or the *alphabet*, of the language. \square

When grammars are defined in such a formal manner, the languages that they produce are called *formal languages*. (It is easy to see that not every sentence in a natural language can be described so formally.)

Let us give a few examples of formal languages.

Example 1.7. Let us define a grammar G as follows:

$$\begin{aligned}
V_T &= \{0, 1\} \\
V_N &= \{\alpha, \beta, S\} \\
S &= \text{start symbol} \\
\Phi : \quad &1. S \longrightarrow 0S\alpha\beta, \quad 2. S \longrightarrow 01\beta, \quad 3. \beta\alpha \longrightarrow \alpha\beta, \\
&4. 1\alpha \longrightarrow 11, \quad 5. 1\beta \longrightarrow 10, \quad 6. 0\beta \longrightarrow 00
\end{aligned}$$

Note: When we write “ $0S\alpha\beta$ ” and so on, it implicitly means “ $0 \circ S \circ \alpha \circ \beta$ ” and so on, where \circ denotes the concatenation operation. Such a simple concatenation operation is often not written explicitly.

It is not difficult to recognize that any string of the form $0^n 1^n 0^n$ can be obtained by using the grammar. We can say that the language $L(G)$ of this grammar is the set $\{0^n 1^n 0^n \mid n \geq 1\}$. \square

Example 1.8. A grammar G is defined in the following manner:

$$V_T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$V_N = \{\text{digit}, S\}$$

$$S = \text{start symbol}$$

$$\Phi : S \longrightarrow \text{digit}$$

$$S \longrightarrow S \circ \text{digit} \quad (\circ \text{ denotes the concatenation operator})$$

$$\text{digit} \longrightarrow 0$$

$$\text{digit} \longrightarrow 1$$

$$\vdots$$

$$\text{digit} \longrightarrow 9$$

Clearly, this grammar defines the nonnegative integers written with possible leading zeros. \square

Sometimes, it may be convenient to use the BNF notational system to define a grammar. The BNF scheme is a slight variation of the notational system that we have used so far. BNF, which stands for the *Backus–Naur form* or the *Backus normal form*, was introduced in the report on ALGOL [78]. The BNF conventions imply that all nonterminals are flanked by wedges $< \dots >$, and the expression $A \longrightarrow B_1 | B_2 | \dots | B_n$ is the shorthand for the set of productions $A \longrightarrow B_1, A \longrightarrow B_2, \dots, A \longrightarrow B_n$. The symbol $::=$ is used for \longrightarrow .

According to the BNF conventions, the grammar of Example 1.8 will be expressed as follows:

$$V_T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$V_N = \{< \text{digit} >, < S >\}$$

$$< S > = \text{start symbol}$$

$$\Phi : < S > ::= < \text{digit} > \mid < S > < \text{digit} >$$

$$< \text{digit} > ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

1.5.2 A Grammar for the Constructive Part of the Metamodel

There is obviously a close resemblance between the constructive part of the metamodel for shape description and the grammar of a formal language. If the metamodel could be expressed as a grammar, it would certainly be simpler to teach it to a machine. But apart from that, we would acquire a more natural vocabulary with which to enquire about a shape description scheme. This is an important point, which we shall clarify shortly.

However, note that in formal language grammar the only operation that is allowed is *concatenation*, which often remains implicit. The crucial point involved in adapting the techniques of formal language theory to shape description is the generalization of the simple concatenation operation to include any complicated shape operations.

A grammar G for the metamodel may take the following form:

V_T : *terminal symbols* that contain two sets –

a set of *primitive shapes* P , and

a set of *shape operators* $*$

V_N : *nonterminal symbols* that could be called $\langle \text{complex_shape} \rangle$

S : *start symbol* that could be called $\langle \text{constructed_shape} \rangle$

Φ : *production rules* (that is, the set C according to our terminology)

$\langle \text{constructed_shape} \rangle ::= P \mid \langle \text{complex_shape} \rangle$

$\langle \text{complex_shape} \rangle ::= [\{P\}, \{*\}] \mid [\langle \text{complex_shape} \rangle, \{*\}]$

$\mid [\langle \text{complex_shape} \rangle, P], \{*\}]$

The notation $[\{X\}, \{*\}]$ means a shape constructed by an operation from the set $\{*\}$ and operands from the set $\{X\}$. In simple words, the production rules Φ specify that any shape that we intend to describe is either a primitive shape or a complex shape. A complex shape is constructed by some operation chosen from the set of shape operators $*$ and the corresponding operands chosen: (a) from the set of primitive shapes P ; (b) from the complex shapes already constructed; or (c) from the combined set of the primitive shapes and the complex shapes that have already been constructed. In principle, a shape operator may be a n -ary operator – that is, an operator that takes a number n of operands – but in practice, mostly unary and binary operators are in use.

1.5.3 An Exploration of Shape Description Schemes in Terms of Formal Language Theory

Some of the concepts of shape description can be reformulated more conveniently by using the vocabulary of formal language.

1.5.3.1 The Description Domain

If a shape description scheme follows a grammar G , then its language $L(G)$ denotes the set of shapes that can all be constructed by the scheme. Therefore, the description domain of the

scheme is a subset of $L(G)$. This means that $L(G)$ gives the upper limit of the description domain.

1.5.3.2 Syntax and Semantics

Two notions are associated with a sentence in any language – the *syntax* of the sentence and the *semantics* of the sentence. The syntax is concerned with the structure of the sentence; that is, how it is constructed by the grammar G . The semantics, on the other hand, is concerned with the meaning of the sentence. Consider the following two sentences: “a student reads the book” and “a book reads the student.” According to English grammar as described in Figure 1.8, these two sentences are both syntactically correct. This means that both of them belong to the language. But while the first sentence carries a meaning for us, the second one does not.

What are the analogous situations in the case of shape description? Here, a syntactically correct sentence means a shape that can be constructed by the grammar of a shape description scheme. But this does not mean that all such constructed shapes will be meaningful to us; only a subset of them may be meaningful. By a “meaningful shape,” we mean a physically valid shape. And just like the semantics of a sentence, the semantics of a shape, as we have already mentioned, is not determined solely by the grammar but, rather, by external references. Consider, for example, the two shapes shown in Figure 1.9.

We can easily imagine a grammar in which both of those two shapes are regarded as syntactically correct shapes. But are both of them meaningful to us? To an engineer, the shape shown in Figure 1.9(b) will certainly be meaningless, since no such object can be realized in practice. However, to a graphic artist the same shape may be perfectly acceptable. Many of the exciting drawings of the Dutch graphic artist M.C. Escher (1902–1972) fall into this category. Thus the notion of physical validity – that is, the notion of the meaning of a shape – is to a large extent application dependent. Detection of the semantic inconsistencies should rely on knowledge of the objects being referred to, and that knowledge may be captured by a set of explicit axioms. A complete shape description scheme, therefore, should consist of not only a grammar G , but also a set of explicit axioms A .

Let us consider another sentence: “heren’t the book is.” According to the same English grammar, this is not syntactically correct, but we can read it as a meaningful sentence.

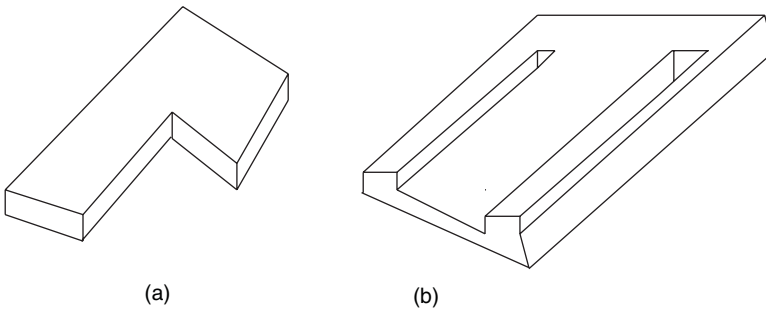


Figure 1.9 Two shapes that may be both syntactically correct, but may not be semantically correct

In the case of shapes, a meaningful but syntactically incorrect sentence means a shape that is perfectly valid according to the notion of the intended application, but that does not belong to the description domain of the scheme that is being used.

1.5.3.3 Ambiguities

Ambiguity is another characteristic of a sentence in a language. A sentence is *syntactically ambiguous* if its grammar makes it possible to write down more than one syntax tree for that sentence. Consider, for example, a sentence from the English language, “they are failing students.” Its syntax trees are given in Figure 1.10.

In the case of natural languages, nonuniqueness of the syntax tree may imply that the sentence has more than one possible meaning. In this particular example, there are two interpretations corresponding to whether “they” refers to a group of students or of instructors.

Similarly, a shape is syntactically ambiguous if it has more than one derivation according to its grammar. In other words, the same shape can be constructed in more than one way within a scheme. In that case, we say that a description scheme does not guarantee *uniqueness* in description. Consider, for example, a shape description scheme that describes shapes as the *swept volume* of a two-dimensional shape translated along some line segment. This means that the scheme considers the line segment and the two-dimensional region as the primitive shapes, and translation as the shape operator. According to this scheme, a rectangular parallelepiped can be described in many different ways (Figure 1.11). A description scheme whose grammar allows such a nonuniqueness description is called a *syntactically ambiguous* scheme.

A sentence may also be *semantically ambiguous*. Even if a sentence is syntactically unambiguous, it may be semantically ambiguous; that is, it may have more than one meaning. The sentence “the astronomer married the star” has two possible interpretations corresponding to two possible meanings of “star.”

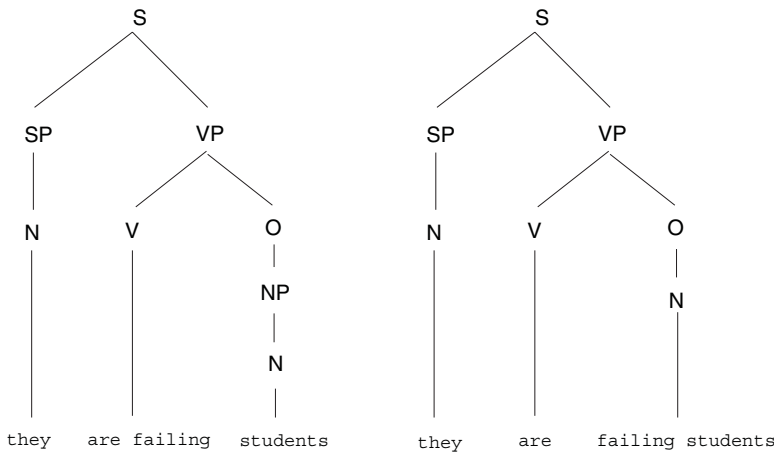


Figure 1.10 The sentence “they are failing students” and its two possible derivations; the grammar used here is slightly different from that in Figure 1.8(a)

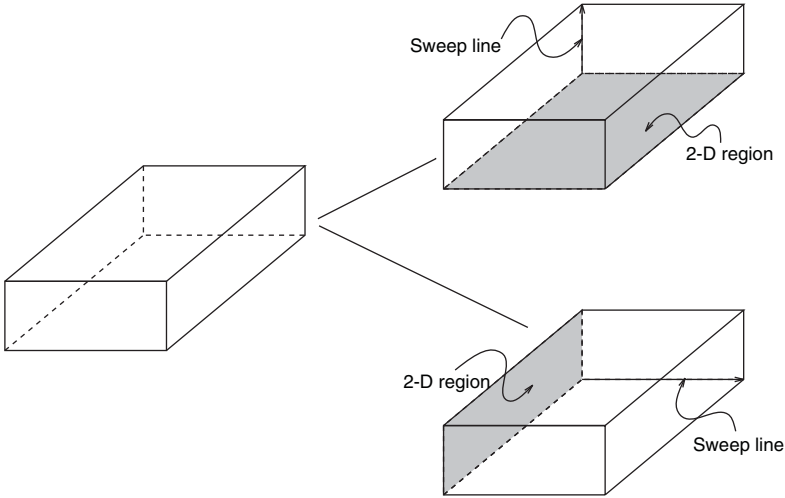


Figure 1.11 An example of nonunique description – two different derivations of a rectangular parallelepiped according to the *swept-volume* scheme

Analogously, a description of a shape is semantically ambiguous if the shape, for a given *canonical derivation*, has more than one meaning or interpretation. We can, for example, take a scheme that includes line segment as the primitive shape and concatenation as the shape operator. It is possible to construct a shape by means of this scheme, like the one as shown in Figure 1.12(a). In Figure 1.12(b), we show how that shape could be interpreted in several ways.

Such semantic ambiguities are frequently encountered in pattern recognition applications. In Figure 1.13, we present an example where the same pattern can be interpreted either as the letter “A” or the letter “H,” depending on the context.

1.5.3.4 Synthesis and Analysis

The notions of shape synthesis and shape analysis can be understood quite clearly in terms of the vocabulary of formal language. The dictionary definitions of these two words go somewhat like this:

- **Shape synthesis/generation:** the combining of separate parts or elements to form a complex whole; production/generation of shapes by systematic application of some rules.
- **Shape analysis:** to separate into its parts in order to identify it or study its structure; given a shape, extraction of the simpler constituents of the shape and relations among those constituents in order to understand the shape.

Through the formal language analogy, these notions can be expressed in more intuitive terms.

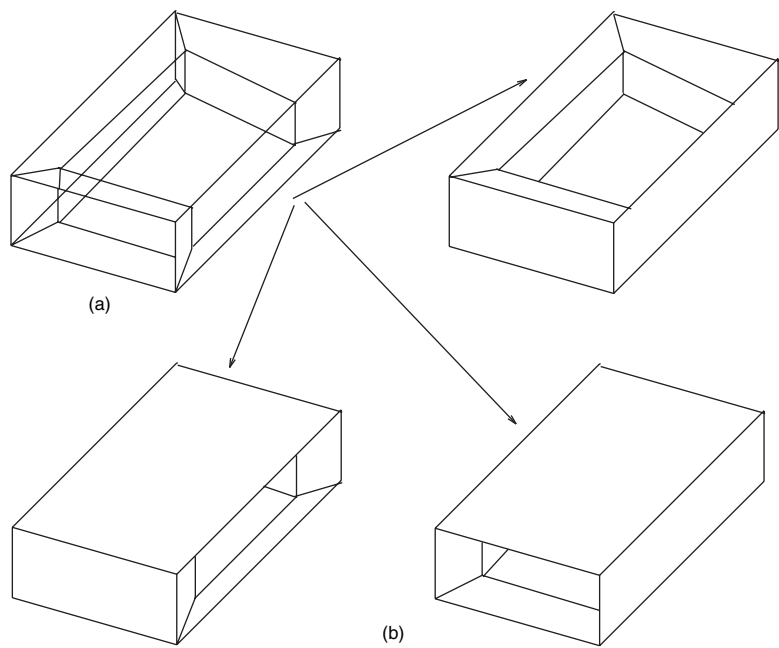


Figure 1.12 An example of semantically ambiguous description – the above shape can be interpreted in many ways

By mere mechanical manipulation of the grammatical rules of some given grammar G , we can *produce* (or synthesize/generate) various sentences in its language $L(G)$. For example, if we use the grammar described in Example 1.7, we can produce any string of the form $0^n 1^n 0^n$. This is equivalent to the shape synthesis process.

In contrast to the generation of a sentence, to *parse* a given sentence is to resolve it into its primitive elements. In parsing a given sentence, we ask (a) whether it is a sentence of the language $L(G)$ of G ? and (b) if so, what its syntax tree is. It is easy to see that the shape analysis task is equivalent to the parsing of a sentence.

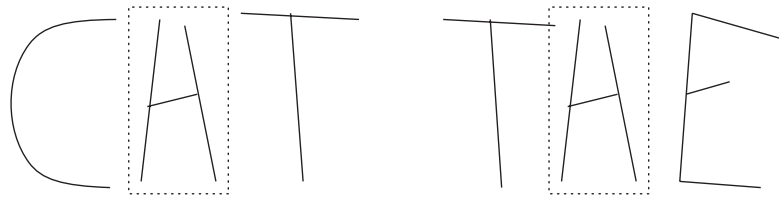


Figure 1.13 An example of semantic ambiguity in character recognition

1.6 The Art of Model Making

1.6.1 What is the Meaning of “Model”?

So far, we have used the term “model” with the assumption that you have an intuitive understanding of its meaning. Let us now make explicit exactly what it means to us. We use “model” in the following sense [74]:

For a given application \mathcal{T} , an object M is a model of an object O to the extent that \mathcal{T} can use M to answer questions about O that are relevant for that application. (By “application,” we mean a set of well-defined tasks.)

Thus the concept of a “model” is inherently a *ternary* relation among \mathcal{T} , O , and M . At no point we can ignore the role of the purpose(s) of the application \mathcal{T} in constructing an M for some O . This is why we emphasized earlier (Section 1.4.2) that every shape description scheme is *purpose-oriented*, and that no scheme is better than all others for all purposes.

The construction of models, as it appears at present, is an art. For any given application, it is possible to design many shape description models. But it requires experience, discernment, and a dash of lateral thinking to design an appropriate one.

The best that we can do is to (a) formulate any shape description scheme within some unified framework (say, within the $(P, *, C, A)$ framework) and (b) assess/evaluate the scheme according to some guideline(s). The big question is: “Is it possible to formulate a reasonably satisfactory set of evaluation guidelines?” At present, we have to answer this question in the negative. However, in the next subsection we will attempt to provide a few such guiding principles.

1.6.2 A Few Guiding Principles

Some guiding criteria have already emerged from our discussions in the previous sections. Consideration of the real world allows us to devise a few more. Clearly, the following list of guiding criteria is far from complete, but it includes the ones that we feel are important for our purpose.

1.6.2.1 Consistency of Axioms

Consistency of axioms is a primary requisite for building up a shape description scheme. (Here, we use the term “axioms” not just in the conventional mathematical sense, but also to mean the production rules C and the explicit axioms A .) To check the consistency of a set of axioms, we have to construct a shape that satisfies all of the axioms. If it is possible to construct any such shape, the set of axioms is consistent.

1.6.2.2 Independence of Axioms

It is only desirable, but not necessary, that the set of axioms should be independent. If we want to check whether or not an axiom, say k , in the set is independent, we have to construct all the shapes that satisfy the other axioms. If all of these shapes satisfy axiom k , then axiom k is dependent; otherwise, it is independent. An alternative approach is to attempt to prove axiom k mathematically from the other axioms.

1.6.2.3 Ease of Identification of the Description Domain

The identification of the system's description domain – that is, the set of shapes that the scheme is capable of describing – is of primary importance. Since almost every shape description scheme that is of any practical use can describe an infinite number of shapes, it is not feasible to enumerate all of them. Therefore, in the pure approach the description domain is specified by stating properties of the shapes in the form of theorems or propositions. For example, the shape domain of our *Toy system* (described in Section 1.4.3) can be completely characterized by stating results such as: “Every element in the *Toy system* space is a bounded or unbounded convex polygon.” In the case of the pragmatic approach, where a set of intended shapes is already given, we try to ensure that the given set becomes a subset of the description domain.

It is often desirable that the description domain should be *large*. In other words, we attempt to capture a space that is as large as possible by choosing as few axioms as are needed. Most of the attempts to *generalize* – from two-dimensional space to n -dimensional space, from the convex domain to the nonconvex domain – fall into the category of enlargement of the shape domain.

1.6.2.4 Uniqueness in Description

Descriptive uniqueness (or syntactic unambiguity) is a highly desirable criterion. If the same shape is described in many different ways in a scheme, it becomes difficult to detect their equivalences by mechanical means. It may be possible to achieve uniqueness by adding a few extra constraints (i.e., by adding more axioms) into the system. Alternatively, a separate procedure could be included that will determine the equality of the shapes described by the scheme.

1.6.2.5 Unambiguity (Mathematical Completeness) in Description

By *unambiguity*, we mean semantic unambiguity; that is, that a described shape can be interpreted in one and only one way. For many applications, the unambiguity of the scheme is a prerequisite. Its necessity can be understood from the example shown in Figure 1.12, which corresponds to more than one object. If our intended application is to render the images of objects after hidden line removal, the description of that shape is certainly not adequate.

In the case of the pragmatic approach, unambiguous description is frequently referred to as *mathematically complete* or *lossless* description. The reason for this is that, given a set of shapes to be described, an unambiguous description contains enough information to distinguish a single object from all other objects in that space and, therefore, is a sufficient source of data for evaluating any mathematically defined function of the object.

However, this does not mean that a mathematically incomplete description is completely useless. It may be adequate for certain applications. For example, consider the application of determining the convex hull of objects. We find that even the description in Figure 1.12 is sufficient for this purpose. The point is that the intended application might not need all of the information on a shape, but only part of it. And if that part of the shape information is unambiguously provided through a description, it is adequate for that application, even if the overall description is ambiguous. For determination of the convex hull, since we only need the extremal points of an object, which are unambiguously provided in Figure 1.12, we do not face any problems.

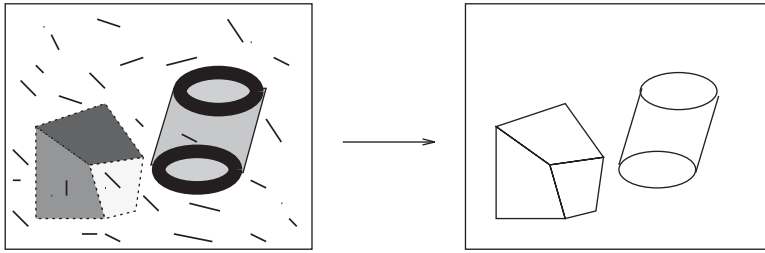


Figure 1.14 An example of mathematically incomplete description in object recognition, which is desirable

In fact, in a number of applications in the image processing and pattern recognition area, a mathematically incomplete description is not only desirable, but indeed required. It is often possible to recognize a complex three-dimensional object from a highly schematized two-dimensional drawing, which implies that only a part of the object's shape information is required for the recognition task. Consider, as a concrete example, the task of recognizing a particular polygon from a given set of polygons. In this case, it is more convenient to represent each polygon by means of the lengths of its edges and the interior angles between adjacent edges (which is an incomplete description, since the position information relating to the polygon in the plane is absent), rather than representing it by specifying the coordinates of its consecutive vertices (which is mathematically complete). In a computer vision task, starting from a complicated image, we may progressively reduce the amount of information in order to arrive at a description that is mathematically incomplete but more ordered and manageable (Figure 1.14). In the case of *lossy image compression* too, we devise description schemes that are mathematically incomplete.

1.6.2.6 Homogeneity in Product and Operand Types

If the task in hand permits, then it is preferred that the product shape is of the same type as the operand shapes. Using the terminology that the operand domain is referred to as the *domain* of operation and the product domain as the *range* of operation, then we can say that the range should preferably be the same as the domain, or a proper subset of the domain.

Note: At this point, you may recall that the evolution of various kinds of number systems has taken place primarily in order to achieve this homogeneity. If we start with the *natural number system* \mathbb{N} and with arithmetic subtraction as the operator, we can achieve homogeneity of operands and products by extending the set \mathbb{N} to the set of integers \mathbb{Z} . The arithmetic division operator in the domain of \mathbb{Z} gives rise to the concept of the set of rational numbers \mathbb{Q} , and so on. On the other hand, to ensure homogeneity in the *integral domain* (i.e., in the domain of integers) with the division operator, the floor function $\lfloor x \rfloor$ (i.e., the greatest integer less than or equal to x) and the ceiling function $\lceil x \rceil$ (i.e., the least integer greater than or equal to x) are introduced.

In the case of shape description, there are many advantages of creating products of the same type as the operands. The primary advantage is that the product shape itself can be used as an operand, without any reformulation. Also, extra validity checking of the product shapes is not needed if all the primitive shapes that are initially chosen are of a valid type.

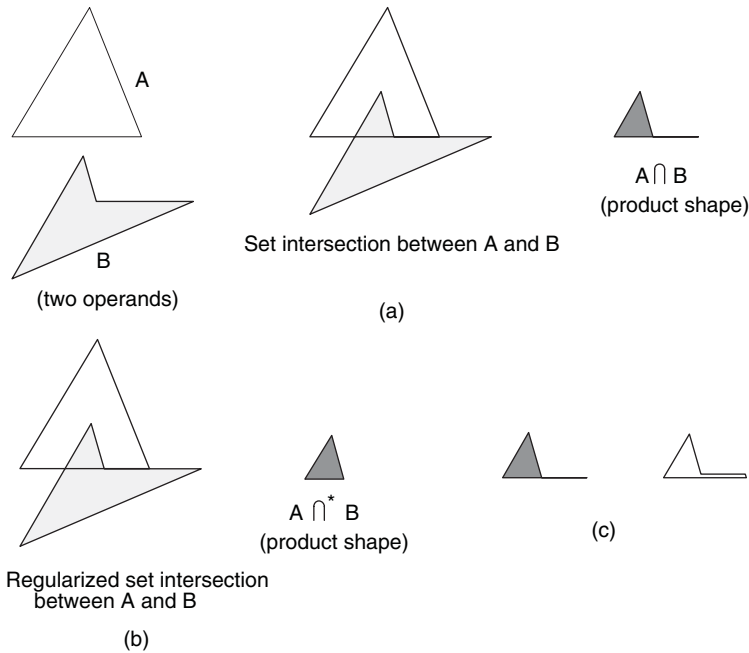


Figure 1.15 (a) The product shape $A \cap B$ is not a two-dimensional polygon; (b) the intersection operation \cap is modified to the “regularized intersection” operation \cap^* to obtain a two-dimensional polygon as a product shape; (c) the definition of a two-dimensional polygon is extended to include degenerate polygons

There are several ways to achieve homogeneity in the type. (a) One approach is to *restrict* the definition of the shape operator. (This is like modifying the division operation to a $\lfloor x \rfloor$ operation or a $\lceil x \rceil$ operation to restrict yourself to the integral domain.) (b) Another approach is to *extend* the definitions of operands and products in such way that both of them can be treated as the same type. (This is like extending the domain from \mathbb{Z} to \mathbb{Q} in order to deal with the division operation.) Consider, for example, a description scheme where the primitive shapes are two-dimensional polygons and the set intersection operator (\cap) is the shape operator. The product shapes are also expected to be only two-dimensional polygons. However, in Figure 1.15(a) we show a situation in which the product shape is not homogeneously two-dimensional; it consists of a two-dimensional polygon together with a one-dimensional line segment as a dangling portion. To achieve homogeneity, we may either: (i) modify the intersection operator to the *regularized intersection* operator \cap^* , which eliminates any such nonhomogeneous dangling portion from the product shape (Figure 1.15(b)); or (ii) extend the definition of the two-dimensional polygon by assuming that the width of a polygon in some direction may be infinitesimally small (Figure 1.15(c)).

1.6.2.7 Scope for Generalization/Extension/Variation

In most of the pure approaches to shape description, it is possible to observe a natural tendency to generalize/extend the schemes. In fact, the tendency is so strong that the area of pure

mathematics is filled with a very large number of generalized spaces – some of which are quite exotic. We can give a few simple examples: two-dimensional Euclidean space, which is generalized to n -dimensional (in fact, *infinite-dimensional*) Euclidean space; the simplest second-degree algebraic equation $x^2 + y^2 = r^2$ (which represents a circle having its center at the coordinate origin), which is readily generalized to the general second-degree equation $ax^2 + by^2 + cz^2 + dxy + eyz + fzx + gx + hy + kz + l = 0$ (which represents a *quadric surface*); the Euclidean *convex* domain, which is generalized to the *generalized convex* domain; and so on.

But why is there a need for generalization or extension of a shape description scheme in the pragmatic approach when the space S to be described is already given? The reason is that although in the pragmatic approach we assume that the description domain S is known, obviously it is never known in a “precise” sense.

Take, for example, an automobile designer trying to design the model of a car. Obviously, at the beginning the shape of the car only exists in the designer’s mind in a fuzzy way. That is, it does not correspond to a single shape, but to a set of shapes that are closely related. In fact, an automobile designer never designs a single vehicle, but several slightly different vehicles, in order to choose the best one from the selection. In designing a telephone handset, various blends (*blends* are the smoothing of intersections of surfaces by introducing a new piece of surface material between the surfaces in question) are always tried out before a particular design is finalized. The situation is not very different for those people who are in the process of analyzing shapes. Only a partial knowledge of the shape domain is available to them. For example, in some computer vision task it may be known that the images involve objects that have only planar faces. Therefore, any shape description scheme used for this purpose must be flexible enough to deal with innumerable scenes that can potentially be created by various combinations of planar-faced objects, and so on.

Let us give two examples to show the types of generalizations that often take place in the pragmatic approach to shape description. In computer graphics and CAD, the concept of a cubic (polynomial of degree 3) *Bezier spline* curve (by means of which we can describe a circular arc only approximately) has gradually been generalized to an n th-degree *rational B-spline* curve, which can provide a single precise mathematical form capable of representing lines, planes, conic curves, free-form curves, and so on. In the image processing area, the use of the classical discrete *Fourier transform* (in terms of a single variable that is expressed as $F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)e^{-j2\pi ux/N}$) has gradually been generalized to a transform of the type $T(u) = \sum_{x=0}^{N-1} f(x)g(x, u)$, where the forward transformation kernel $e^{-j2\pi ux/N}$ of discrete Fourier transform is replaced by a general function $g(x, u)$. As a result, several transforms – including the Fourier, Walsh, Hadamard, discrete cosine, Haar, and Slant transforms – can be obtained from a single framework.

Although the effort to generalize may give rise to an elegant and compact description scheme, the associated problem is the increased complexity of the scheme. The more general a description system becomes, the more difficult it is to work out its mathematical characteristics. To clarify this point, we will present a simple example.

Assume that our task is to devise a description scheme for a given set of *regular convex* polygonal regions, as shown in Figure 1.16(a). For regular convex polygons, as is well known, there are quite a few elegant schemes in existence. One such scheme is to set up a Cartesian coordinate system at the center O of a regular p -polygon (where p denotes the number of edges/vertices) in such a way that the Cartesian coordinates of the vertices of the polygon can

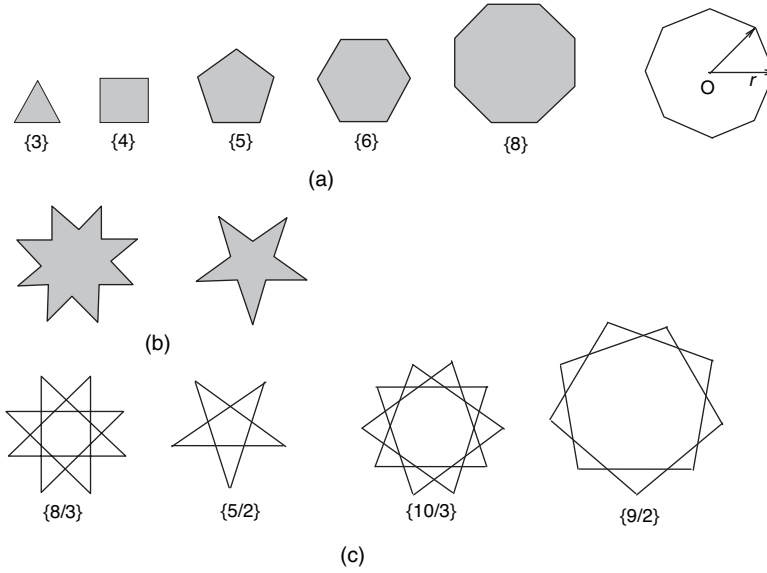


Figure 1.16 The generalization of a shape description scheme from *regular convex* polygonal regions to *regular star-shaped* polygonal boundaries: (a) regular polygons; (b) two star-shaped polygons; (c) a generalized description of regular and star-shaped polygons

be represented as follows:

$$\left(r \cdot \cos \frac{2k\pi}{p}, r \cdot \sin \frac{2k\pi}{p} \right), \quad k = 0, 1, \dots, p-1, \quad (1.9)$$

where r denotes the distance of O from any of the vertices (as shown in the bottom diagram of Figure 1.16(a)).

If two regular *star-shaped* polygonal regions (shown in Figure 1.16(b)) are now added to the list of the set of given shapes, it is no longer possible to use the same description scheme. At this point, we feel the need for a more generalized scheme to describe regular polygons – convex as well as star-shaped within a single framework. An interesting generalization is possible if we decide to describe any given polygonal region S_p in terms of its boundary ∂S_p , which is allowed to be *self-crossing*. That, in turn, compels us to introduce a more complex shape operator ∂^{-1} in our description scheme in order to obtain S_p from ∂S_p , which may be self-crossing, in general. Our generalized scheme can be expressed as follows:

$$S_p ::= \partial^{-1}(\partial S_p), \quad (1.10)$$

$$\partial S_p ::= \text{POLYLINE}(a_0, a_1, \dots, a_m = a_0), \quad (1.11)$$

$$a_k ::= \text{ROTATION}_{2k\pi/p}(a_0), \quad (k = 0, 1, \dots, m), \quad (1.12)$$

where p is a rational number of the form m/d_p .

The scheme has only one primitive shape, a point in the plane whose position vector is denoted by a_0 . There are three shape operators: ∂^{-1} , *POLYLINE*, and *ROTATION* $_{2k\pi/p}$. *POLYLINE* $_{(a_0, \dots, a_m)}$ denotes the joining of the ordered set of points a_0 to a_m by line segments, and *ROTATION* $_{2k\pi/p}$ denotes the rotation of the point a_0 by an angle $2k\pi/p$ around the origin, where the value of p is fixed for each polygon S_p .

It is easy to see that when p is an integer (i.e., $d_p = 1$), ∂S_p becomes a regular convex polygon. The complexity of the system arises because of the generality of p ; that is, when p is not an integer and as a consequence ∂S_p becomes self-crossing. For some noninteger p 's, the corresponding ∂S_p 's are shown in Figure 1.16(c).

1.6.2.8 The Limitations of Construction Tools and Approximation

The limitations of the available construction tools – theoretical as well as physical – quite often compel us to resort to *approximate* description schemes. Such limitations may either be absolute in nature or imposed by some practical considerations. When we say that “this application does not call for absolute accuracy,” we often mean that some part of the accuracy can be sacrificed in consideration of the limited amount of resources available at hand. We can briefly mention some of the factors that are responsible for giving rise to approximate models.

- *The absence of an accurate solution.* There are certain problems whose accurate solution is not possible even in principle. Consider, for example, problems such as finding the length of the edge of a square whose area is two units, or determining the area of a circle whose radius is one unit. Because the solutions involve, respectively, the computation of $\sqrt{2}$ and π (πi) values, which are both irrational numbers, in such situations there is no option but to accept approximate solutions. In most cases, it may not do any harm if we accept the value of π as 3.1416, or even better as 3.14159, and things will certainly be much better if we accept the value as 3.1415926535897932384626433832795, but all these are still approximate. In the same light, it appears that the exact descriptions of shapes of natural objects, such as mountains, clouds or trees, are seemingly impossible. But approximate models of such shapes often serve our purpose well.
- *Dependence on infinite numerical algorithms.* Computations in geometry are not entirely *combinatorial* (such as, given n points in the plane and a query point x , report the point closest to x), but also *numerical* (such as, find the intersection points between two given curves). One primary problem with most of the numerical algorithms (an “algorithm” means an effective procedure; i.e., a finite set of rules that gives a sequence of well-defined operations for solving a problem) is that the algorithms are *infinite* in nature. By “infinite algorithms” we mean those algorithms that give better and better estimates of the results the longer the algorithms continue. For example, consider the problem of determining the root of the function $f(x) = x^2 - 9 = 0$. The Newton method for solving $f(x) = 0$ is given by

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i = 0, 1, 2, \dots, \infty, \quad (1.13)$$

where x_i denotes the value of the root at the i th estimate, x_0 is taken as the initial approximation, and $f'(x) = df/dx$. If we start with an initial approximation $x_0 = 9$, we obtain successive values of x_i as 9, 5, 3.4, 3.02353, ... However, within a finite time, such an

algorithm cannot be guaranteed to generate the exact solution. It is only possible to ensure that the approximate result will be within a given neighborhood ϵ of the exact solution; that is, $|x_i - x_{exact}| \leq \epsilon$. To get a better idea of this kind of approximation, we invite interested readers to consult books on numerical analysis.

- *The limited numerical precision of the digital computer.* An exact solution of a numerical problem means computing the solution with infinite precision. Clearly, a computer that has only finite resources cannot provide such a precision. Therefore, a computer cannot use the real number system, but only simulates it by means of a system called the *floating-point* number system. There are many good texts on the floating-point number system [53] and we do not intend to discuss it here. But we must point out that the floating-point number system is not a proper model of the real number system. In other words, floating-point computation is by nature inexact, and as a consequence several fundamental notions of real number arithmetic, on which every shape description scheme is based, is no longer remain valid. For example, the associative and distributive laws for addition and multiplication are no longer true in floating-point arithmetic; that is,

$$a(b + c) \neq ab + ac, \quad (1.14)$$

$$a(bc) \neq (ab)c, \quad (1.15)$$

$$(a + b) + c \neq a + (b + c), \quad (1.16)$$

where a , b , and c are floating-point numbers.

The effects of the collapse of these fundamental laws are severely felt in geometric computations. For example, it is possible to get into the following situation: we find that a line segment from point a to point b intersects a line segment from c to d , but by reversing the segment directions the segment from b to a does not intersect the segment d to c . Many such numerical problems in geometric computations are presented in Forrest [19,20].

- *Limited resources on hand.* Even if it becomes possible to obtain an accurate solution, the limited availability of resources may force us to adopt an approximate solution. For example, the exact descriptions of quadric surfaces (i.e., the surfaces of spheres, cylinders, cones, ellipsoids, paraboloids, and hyperboloids) are known. But in the CAD modeling of quadric surfaces we often approximate them by planar surfaces to reduce the computational complexities in manipulating such surfaces (Figure 1.17).

We must note that, by definition, an approximate model is a mathematically incomplete model. This means that the problem of semantic ambiguity can often arise. Two slightly different cylinders, for example, may have the same description in terms of planar surfaces. In this context, the question of *stability versus sensitivity* should be properly addressed. By “stability,” we mean that if two shapes are similar, then their approximate descriptions must reflect that similarity, while “sensitivity” means that even subtle differences between two shapes must be expressible. These opposing conditions can be satisfied only if it is possible to decouple stable information that captures the more general and less varying properties of a shape from information that is sensitive to the finer distinctions between shapes. The notion of *multi-scale* representation [18,58] is one step in that direction.

In summary, we can say that a truly accurate description of a set of shapes may not be attainable at times, or sometimes it may not even be desired. But in order to understand and to fully exploit an approximate model, the mathematics of shape description must require a theory

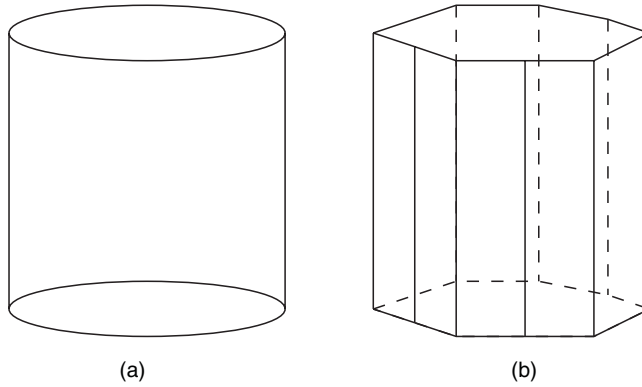


Figure 1.17 (a) An example of a quadric surface – a cylindrical surface; (b) approximation of the cylindrical surface by a mesh of polygons

of approximation that will deal with topics such as the transformations of the mathematical laws in approximate systems, the measure of approximation in every situation, the relationship between computational complexity and the approximate model, and so on. However, such a full-fledged theory is yet to be devised.

1.6.2.9 The Validity of Shapes in the Context of Application

For a given application, we must precisely formulate what the mathematical properties are that a valid (“valid” according to the notion of that application) shape must satisfy, and then include a suitable set of axioms in the description scheme to ensure such validity. The inclusion of the axioms could be done either (a) by adding a few more explicit axioms to the set A , or (b) by modifying/selecting the primitive shapes P , the shape operators $*$, and the production rules C in such a way that every constructed shape becomes valid. Although the former approach may be relatively easier to incorporate, the latter one may become advantageous in the long run; if there are no explicit axioms in a scheme, no extra validity checking is needed.

1.6.2.10 Conciseness in Description

Conciseness refers to the amount of data required to describe the shape of an object in a description scheme. In general, in the case of the pure approach, no *redundancy* is permitted in describing shapes; the equation of a curve or the specification of a convex object by means of supporting hyperplanes do not contain any redundant information. That may not, however, be true in a pragmatic approach, and thus the question of conciseness arises.

It may appear at first sight that most shape description designers will argue in favor of conciseness; the more concise a model is, the more convenient it is to store it and to transmit the data.

However, the necessity of conciseness cannot be precisely established. First, the requirement for conciseness frequently conflicts with the requirement for simplicity in description. With

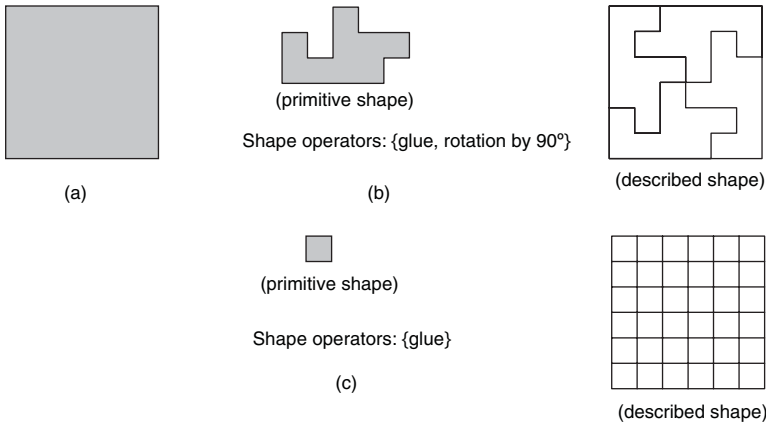


Figure 1.18 A demonstration of conciseness versus simplicity: (a) a 6×6 unit square that is to be described; (b) a concise description (scheme I) using complex primitive shape and shape operators; (c) a verbose description (scheme II) in which the primitive shape and shape operator are simple (the *glue* operation is like concatenation; it is a restricted form of set union that applies only to shapes with disjoint interiors)

more complex primitive shapes and shape operators, it is possible to obtain a more concise description. Take, for example, the description of a 6×6 unit square (Figure 1.18(a)) in two different schemes. In the first scheme (Figure 1.18(b)), the primitive shape and the shape operators are relatively more complex than those in the second scheme (Figure 1.18(c)). But the description of the square is more concise in the first scheme than in the second one.

In the case of shape description involving the computer, concise description gives rise to a few other problems. For example, concise description reduces the storage space at the expense of more computation. Thus the computational efficiency may be improved considerably if we use a verbose description that depends more on storing the data than on actual computing.

Example 1.9. The following example, though artificial, may be illustrative at this point. Assume that the shape of some object involves ten discrete points on a one-dimensional line. According to some conveniently chosen coordinate system, that shape can be described as a set: $S = \{1, 2, 3, 5, 8, 13, 21, 34, 55, 89\}$. This particular set can also be described more concisely as $S = \{p_i \mid p_1 = 1, p_2 = 2, p_{i+2} = p_{i+1} + p_i \text{ for } i = 1, 2, \dots, 10\}$, since the numbers involved are Fibonacci numbers. Now questions such as "Does a given point x belong to the set?" or "What is the distance between the fourth and the eighth point of the set?" can be answered more quickly from the former description than from the latter one. Imagine the case of $i = 1, 2, \dots, 100$. □

Unfortunately, relatively few formal tools are available at present to judge quantitatively the relation between conciseness and computational complexity. In particular, there is a serious need for a theory in the direction of *space-complexity tradeoffs*.

1.6.2.11 Efficacy in the Context of Applications

We have emphasized several times that every shape description scheme should be application oriented, and that the efficacy of a scheme must be judged in the context of its application.

Any application task \mathcal{T} can be viewed as a two-part procedure. The kernel part can be thought of as an algorithm *kernel* that accepts certain information about a shape, or shapes, as its input data D_{in} and produces the desired output data D_{out} . Now, all of the relevant information about the shape(s) is supposed to be contained in the description, say M , within the scheme. Therefore, the second part of \mathcal{T} should be another algorithm that accepts M as its input data and should produce as output the information D_{in} that is needed for *kernel*. Let this algorithm be called *extract*. The application task \mathcal{T} may, therefore, be expressed as follows:

$$\begin{aligned}\mathcal{T}: \text{extract}(M) &\longrightarrow D_{in}, \\ \text{kernel}(D_{in}) &\longrightarrow D_{out}.\end{aligned}$$

It is obvious from the above formulation that the efficacy of a description scheme in the context of \mathcal{T} will depend on how efficient the algorithm *extract* can be made.

If the information D_{in} cannot be extracted from M at all, we say that the description is inadequate for the task \mathcal{T} . For a mathematically complete description scheme, such a situation will never arise. But even though it is possible to extract D_{in} from M , the algorithm *extract* becomes harder if D_{in} is not explicitly available in M . We have already touched upon this point in Section 1.4.1.

To date, little is known about the quantitative measure of the efficiency of *extract* for some given M and \mathcal{T} . The type of problem that we face can be demonstrated by means of an example. The application task \mathcal{T} , we consider, is to determine the intersection of two given geometric surfaces. Any intersection algorithm essentially needs all of the points that constitute each of the given surfaces as D_{in} and produces as output D_{out} the points that are common to both of the surfaces. In the digital domain (where the objects are described on a digital grid, which is either two-dimensional or three-dimensional), the *spatial occupancy array* (i.e., grid point by grid point) description, therefore, appears to be most suitable for this purpose. But even then, since this kind of description is very verbose, surfaces are generally described by means of analytical equations. An analytical equation may take various forms; for example, an *explicit* form, an *implicit* form, or a *parametric* form. Can we now say which of these three forms will be most suitable for intersection? The answer, that the intersection problem is simplified if one of the surfaces is defined implicitly and the other one is defined parametrically, is not obvious. Even if we assume that this fact is known, can we decide which of the two surfaces should be defined implicitly and which one parametrically in order to obtain greater efficiency? To work it out, let us begin with the implicit equation of the circle and the parametric equation of the line; that is,

$$\text{Circle : } (x - x_{center})^2 + (y - y_{center})^2 - r^2 = 0, \quad (1.17)$$

$$\text{Line : } x = x_0 + at, \quad y = y_0 + bt. \quad (1.18)$$

Substituting the parametric line equations for x and y in the circle equation, we obtain

$$(x_0 + at - x_{center})^2 + (y_0 + bt - y_{center})^2 - r^2 = 0, \quad (1.19)$$

which is quadratic in t , and therefore not difficult to solve.

Conversely, if we describe the circle by a parametric equation and the line by an implicit equation; that is,

$$\text{Circle : } x = x_{center} + r \cos \theta, \quad y = y_{center} + r \sin \theta, \quad (1.20)$$

$$\text{Line : } bx - ay + (ay_0 - bx_0) = 0, \quad (1.21)$$

then the following trigonometric equation needs to be solved:

$$b(x_{center} + r \cos \theta) - a(y_{center} + r \sin \theta) + (ay_0 - bx_0) = 0. \quad (1.22)$$

This equation is obviously more difficult to solve than the previous one.

However, we should remember that a shape description scheme is not designed for a single application task, but for variety of applications. The requirements for some of these applications might turn out to be conflicting. For example, while description of a shape in the form of a *spatial occupancy array* is preferable for the purpose of intersection, it is not a good choice at all for applications where boundary information about the shape is explicitly needed. Moreover, the range of applications may not be known precisely at the time of designing/choosing a description scheme.

1.6.2.12 Simplicity in Description

In order to minimize the effort during interaction with a described shape, a shape description scheme must be simple. However, the notion of simplicity is somewhat vague, since it can be viewed from different angles. For example, by “simple” we may mean that the description of a shape should be sufficiently simple to be conceptualized by a human user, or we may mean that the description should be simple enough to be manipulated by the theoretical tools, or that it should be simple enough to be manipulated by a machine, and so on. Now *conceptual simplicity* partly turns out to be subjective, since it is certainly a function of familiarity: the more familiar a user is with a system, the closer the description comes to his or her mental schema. Similarly, *computational complexity* (the inverse of simplicity) cannot be precisely defined.

No area of computer science and computational mathematics is as replete with such a variety of seemingly disjoint notions as the study of computational complexity.

We can point out in this context that, in the past, most of the familiar description systems, particularly in the pure approach, have aimed to achieve conceptual simplicity. As a result, the primitive shape elements P are frequently chosen to be either points or straight lines. The exceptions – such as the Lie’s *sphere-geometry* (where the primitive shapes are spheres) [22], Pedoe’s *algebra of circles* (where the primitive elements are circles) [79], Grassmann and Plücker’s idea of generalized shape elements [44], and so on – are very few in number. Also, the set of axioms R or shape operators $*$ is kept at a conceptually simple level. That is understandable, considering the fact that the explorations of the description schemes were mostly carried out by human hand.

However, in recent times, when shape manipulations are frequently being carried out entirely by computer, computational simplicity is becoming more significant than conceptual simplicity. The description of a gray-scale image by means of *sin* and *cos* functions (i.e., the *Fourier transform*) is almost impossible to carry out manually, but it is a trivial task for a machine

that uses some efficient FFT algorithm. This change of perspective is, in fact, bringing about significant changes in devising shape description schemes.

1.7 Shape Description Schematics and the Tools of Mathematics

The discussions so far have hopefully made it clear that to devise/evaluate a shape description scheme we need a variety of mathematical tools, from seemingly different fields. First, a thorough knowledge of various geometric shapes (convex shapes, polytopes, regular shapes, simply connected shapes, curves and surfaces, etc.) and various geometric operators (set operations such as union, intersection, and difference, geometric transformations such as rotation, scaling, and shearing, convolutions, various functions including recursion, vector operations, etc.) is required in order to choose/identify appropriate primitive shapes and shape operators for the scheme. Second, to assess the mathematical characteristics of the scheme in accordance with the guiding criteria given above, concepts such as the equivalent relation, the bounded set, the closed set, connectedness, the boundary of a set, and so on must be known. Third, in order to carry out various operations on shapes (such as synthesis of shapes, modifications of already specified descriptions, rendering/display of shapes, transformations of various types, decomposition/analysis of the given shapes, etc.) with the help of a computer, we need several computational tools, including algorithm design tools, data structure design tools, numerical methods, and suchlike, along with tools for measuring the computational complexities of algorithms, for understanding floating-point arithmetic, and so on. Clearly, this is only a partial list of the mathematical tools that are needed.

To date, the usual method of studying these tools is very haphazard. The tools are used whenever they are needed, without ever establishing the rationale behind their use. By “establishing the rationale” we mean establishing the logical basis for using a tool for a certain purpose, in order to justify logically why this tool, rather than some other tool, is most appropriate for the purpose.

As a result, at present, the discipline of shape description does not have a *monolithic* structure; it contains widely scattered mathematical techniques – disparate and at first sight, unrelated. There is another problem that, we consider, is more serious, though less apparent. As we have already mentioned (Section 1.4.1), the use of a mathematical system to solve a practical problem is simply a *mapping* of the relevant part of the problem to the relevant part of the mathematical system. It is always implicitly assumed that the characteristics of the problem, the basic axioms of the mathematical system, and the nature of the mapping function are known “precisely.” However, this is often not the case. Quite frequently, we use a mathematical tool without being at all aware of the assumptions behind such a mapping. As a result, it becomes difficult to achieve anything other than imitation, and at times things may even go wrong. To clarify this point, we will give a few simple examples in the next section.

1.7.1 Underlying Assumptions when Mapping from the Real World to a Mathematical System

Here, we shall consider some frequently used mapping functions and examine some of their underlying assumptions.

1.7.1.1 Coordinatization of Points

To process any geometric object by means of a computer, the first task is to *coordinatize* the relevant points of the objects. The Cartesian coordinate system is mostly used for this purpose, although it is just one of the many methods of coordinatization. We briefly look into some of the basic assumptions behind this frequently used mapping function and highlight them.

If the geometric object of our concern lies in an n -dimensional space (n being 2 or 3 in the ordinary real world), for coordinatization we need to introduce $n + 1$ items: one point o , called the origin; and n basis vectors $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_n\}$.

- The set of items $\{o, \vec{u}_1, \vec{u}_2, \dots, \vec{u}_n\}$ is an imposition that is completely external to the geometric object, and the choice of the set is quite arbitrary.

Any point p and any vector \vec{v} in the n -dimensional space can be uniquely represented as $p = o + \alpha_1 \vec{u}_1 + \dots + \alpha_n \vec{u}_n$ and $\vec{v} = \beta_1 \vec{u}_1 + \dots + \beta_n \vec{u}_n$, where α_i, β_i , and so on are scalar/real numbers. The sets of scalars $(\alpha_1, \dots, \alpha_n)$ and $(\beta_1, \dots, \beta_n)$ are called the coordinates of p and \vec{v} , respectively.

- Although the coordinate representations of a point p as well as a vector \vec{v} appear to take exactly the same form, the position of the origin o is the additional factor that is implicitly required in the case of a point.

Although for coordinatization it is not necessary that the basis vectors must be orthonormal, it is convenient to use an orthonormal basis, which gives rise to the Cartesian coordinate system.

- The Cartesian coordinate system is a special class of coordinate system where the basis vectors are orthonormal; that is,

$$\vec{u}_i \cdot \vec{u}_j = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases} \quad (1.23)$$

The barycentric coordinate system offers an alternative method of introducing coordinates. Here, we use an n -simplex, which is simply a collection of $n + 1$ points that are in the general positions (i.e., the points are such that none of them can be expressed as an affine combination of the others). The 1-simplex is a line segment, the 2-simplex is a triangle, the 3-simplex is a tetrahedron, and so on.

- An alternative is the barycentric coordinate system, in which the coordinates of a point p are $(\alpha_0, \alpha_1, \dots, \alpha_n)$, where $\alpha_0 + \alpha_1 + \dots + \alpha_n = 1$, and that of a vector \vec{v} is $(\beta_0, \beta_1, \dots, \beta_n)$, where $\beta_0 + \beta_1 + \dots + \beta_n = 0$.

There is no doubt that quite often we use a coordinate system without being aware of these facts.

1.7.1.2 Geometric Transformation by Matrix Multiplication

Geometric transformations of points, for the convenience of machine computation, are generally formulated in terms of matrix multiplication. Consider, for example, the two-dimensional counterclockwise rotation of the point (x, y) onto (x', y') by angle θ , which can be expressed in matrix form as $[X'] = [T][X]$; that is, as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (1.24)$$

In order to rotate digital pictures by maintaining the *scanline order* for raster-scan output devices [83], the transformation matrix $[T]$ is decomposed as a product $[T] = [T_1][T_2]$:

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \tan \theta & \sec \theta \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ 0 & 1 \end{bmatrix}. \quad (1.25)$$

What is the geometric interpretation of matrices $[T_1]$ and $[T_2]$? It is not immediately obvious that $[T_1]$ is a combination of *shearing and scaling* transformations in the y -direction, and that $[T_2]$ is the same in the x -direction. It is also not easy to guess what problems might arise if we were to view a rotation transformation as a product of combinations of shearing and scaling transformations.

1.7.2 Fundamental Mathematical Structures and Their Various Compositions

In this book, instead of surveying the relevant mathematical tools as disparate techniques, we attempt to unify them under some central theme. In our development of the metamodel, such a unifying viewpoint presents itself quite naturally: it is the pursuit of *structure*.

The basic raw material of any mathematical study is the concept of a *set* – a set of natural numbers, a set of integers, set of points, and so on. But a set alone is a completely amorphous, structureless object. Therefore, sets are given various *structures*; that is, various conditions, which should be satisfied at all times, are imposed among the elements of a set. A set endowed with some structure is called a *system* or a *space*.

It has already been mentioned that any shape description scheme is a system or space S whose structure is defined by P and R . To devise/evaluate a description scheme essentially means to uncover that structure. But the structure of S is a *composite* structure and is often very complex to explore. Most often, the primitive shapes P are sets that already possess some structure. For instance, consider primitive shapes such as the line segment, the square region, or the rectangular box. All of these are sets of points endowed with some strong structure. When such shapes are combined following the axioms in R , more structure is imposed on them. Can we predict the possible structures associated with these shapes in S -space?

Mathematicians claim that, at the present time, the human mind can only conceive of the following types of structures:

- (a) Algebraic structure.
- (b) Analytical structure:

- (i) topological structure;
- (ii) measure structure.

Any other structure is a composite of the above structures.

Let us briefly review what each of those structures means. *Algebraic structures* in a set are essentially those that permit the “composition” of two or more elements (in special cases, only one element), leading to another element of the set. We all are familiar with various algebraic structures. A trivial example is the set of all real numbers \mathbb{R} , with ordinary addition “+” as the composition operator (also called the composition law). Thus $(\mathbb{R}, +)$ is an algebraic system. Similarly, (\mathbb{R}, \cdot) or $(\mathbb{R}, +, \cdot)$ are also algebraic systems, where “ \cdot ” represents ordinary multiplication. A nontrivial example may be the algebraic system (A, \square) , where the set $A = \{a, b\}$ and the composition operator \square is specified by the following table:

\square	a	b
a	a	a
b	a	b

The table expresses the fact that, in the system, (A, \square) , $a\square b = a$, and so on.

Most geometric objects possess algebraic structures. Consider, for example, some geometric object – say, a line segment. This is a set of points combined in a certain way so as to obtain all points on the line segment. The composition law for a line segment through the points a and b states that $\lambda a + \mu b$ is a point on the line segment, provided that $\lambda + \mu = 1$ and $\lambda, \mu \in \mathbb{R}$, where \mathbb{R} is the set of all real numbers. A set of points can be combined in several other ways to obtain more complex curves.

Analytical structures have two major subclasses. In the case of *topological structures*, we are concerned with relations between elements (of a set) that can somehow be characterized by the concept of “neighborhoods.” Take a line segment as an example again. We can note that points on the line have neighborhoods, and that those neighborhoods are related to intervals. We may also become aware of properties such as the continuity or boundedness of a line segment. If there is more than one line segment, then we may be concerned as to whether they are connected or disconnected, and so on. All of these properties can be defined in terms of neighborhoods. Such properties of the elements of a set are considered in the study of topological structures.

The second subclass of analytical structures, which is called *measure structures*, is used to formulate the notion of “extent” (length, area, volume, mass, etc.). Apart from the algebraic and topological structures of a line segment, we also talk about the length of a line segment. Proceeding from the line to the plane and to three-dimensional space, the concepts of area and volume reveal themselves to us. During our experiences with material bodies, we, furthermore, become conscious of another kind of extent or measure, such as the mass of the object.

Note: The notions of distance, size of angle, and so on are not taken into consideration in Euclidean geometry. Several years after Euclid’s work, the famous Syracusan mathematician Archimedes supplemented Euclid’s system of axioms with a further system of five axioms, which he needed in connection with investigations into the lengths of curves, the areas of surfaces, and the volumes of solids [9].

The following points should be noted in connection with the structure of a set:

- A given set may possess more than one kind of structure. In fact, much of the work of contemporary mathematics consists precisely of interrelating various structures in an intimate and mutually interdependent manner. *Functional analysis*, for example, superimposes topological and/or measure structures on algebraic systems. In *analytic geometry*, the algebraic structure and measure structure of geometric objects are studied in a mutually interdependent manner.
- It often happens that two seemingly different structures may exhibit similar characteristics and, in fact, one structure may be used to study the other. This is the notion of *homomorphism*, or simply *morphism* between two structures. There are many examples of morphism. A simple example of morphism between two algebraic systems is the following. Consider (\mathbb{R}^+, \cdot) – that is, the set of all positive nonzero real numbers equipped with ordinary multiplication – and $(\mathbb{R}, +)$, the set of all real numbers with ordinary addition as the composition operator. These two systems are homomorphic, since we can define a mapping function

$$\log : (\mathbb{R}^+, \cdot) \longrightarrow (\mathbb{R}, +)$$

between the two systems. We know that for every $a, b \in \mathbb{R}^+$, $\log(a \cdot b) = \log a + \log b$.

The idea of morphism plays a central role in the study of geometric problems by means of algebraic systems. We all are aware of the correspondences between geometric objects – such as points, lines, and curves – and algebraic objects, such as numbers, sequences of numbers, and functions on numbers. This is the basic idea behind analytic geometry, which was introduced by the French mathematician and philosopher René Descartes (1596–1650).

Our survey of mathematical tools will, therefore, be the exploration of the relevant structures. In the next chapter, we will start with an introduction to *sets*. We will then consider *algebraic structures* and *topological structures*. However, the study of those two structures will be by no means comprehensive; we have decided to present only a few selected algebraic and topological systems that have direct application to shape description. On the question of *measure structures*, we do not follow the same course of study. The independent establishment of the general notions of measure structure is both more subtle and more complicated than the foundation-laying of algebraic and topological structures. Therefore, we have decided to be more specific and consider *analytic geometry*, where measure structures of geometric objects are explored interdependently with their algebraic structures.

The other necessary tools, such as general tools for designing and analyzing computer algorithms, are not discussed in this book; nor are the general methods of numerical algorithms covered. Once again, we want to be more specific on these issues and, therefore, we will concentrate on those geometric algorithms that have a direct bearing on shape description problems.

2

Sets and Functions for Shape Description

2.1 Basic Concepts of Sets

2.1.1 Definition of Sets

The concept of a set is the most general and most fundamental notion of mathematics. Precisely because of this utmost generality, it is extremely difficult to give a logically unassailable definition of a set. Informally, by a *set* we mean a collection of objects of any sort. The term “object” is used here in a very broad sense, to include even abstract objects. Thus, a pair of shoes, the set of all Indians, the set of all shape description schemes, the set of points in a line segment, the set of all triangles in the plane, and the set of all ideas contained in this book are all examples of sets.

The apparent simplicity of this notion is so deceptive that a careless use of the above definition can quickly lead to logical paradoxes. A bitter battle was fought between mathematicians for many decades, with the aim of eliminating these enormous difficulties. However, it is not within our scope to even touch upon those issues. For more details, see any of the standard books on the foundations of mathematics, such as [10,17,57].

Fortunately, we can circumvent the pitfalls by a simple consideration. All the sets that we are going to consider will always form “parts” of some “bigger” sets, whose existence we merely postulate. In every specific study, we assume – either explicitly or implicitly – the existence of some *universal set* S , and all sets A, B, \dots, X, Y, \dots that we consider will be collections of objects that are taken from the universal set S . For example, when we talk about the set of all triangles in the plane, our stipulated universal set S is the set of all points of the plane. Thus S is our *universe of discourse*.

Note that we encounter many words that convey the same idea as that of a set. The terms “class,” “aggregate,” and “collection” are often used as synonyms of the term “set,” particularly to avoid using the same word repeatedly in a given sentence. For example, a set of sets can be called a *collection* of sets.

2.1.2 Membership

A fundamental concept of set theory is that of membership, or belonging to a set. Any object belonging to a set is called a *member* or an *element* of that set. Generally, we denote sets by capital letters and members or elements by lowercase letters. If x is an element of set X , we write $x \in X$. The notation $x \notin X$, on the contrary, means that x is not a member of X . If several elements of the same set are considered, we distinguish them by subscripts or superscripts (enclosed in parentheses).

2.1.3 Specifications for a Set to Describe Shapes

A set is considered to be known if we know what its elements are – or at any rate if, in theory, we can find out. There are many ways of specifying a set.

1. The simplest way of specifying a set is to list all its members. The standard notion is to enclose the list in curly brackets. For example,

$$A = \{0, 1, 2, \dots, n\} \quad (2.1)$$

denotes the set of the first n natural numbers. Similarly,

$$B = \{\text{spring, summer, fall, winter}\} \quad (2.2)$$

denotes the set of seasons.

If it is impossible to “write down” all of the elements, conventions and habit often permit the use of the following notation. For example,

$$\mathbb{N} = \{0, 1, 2, \dots\} \quad (2.3)$$

clearly indicates the set of all *natural numbers*.

2. Instead of a list, we can give a property that specifies precisely the elements that we wish to include in the set. Let S be a given universal set and let \mathbf{P} denote some “recognizable” property that some elements of S might have. Then we can specify a set X by collecting all elements of S that share the property \mathbf{P} . (We can also say that we collect those elements of S for which the proposition \mathbf{P} is true.) We then write

$$X = \{x \mid x \in S \text{ and } x \text{ has the property } \mathbf{P}\}. \quad (2.4)$$

Since the existence and nature of the universal set is usually obvious, we often abbreviate the notation and simply write

$$X = \{x \mid x \text{ has the property } \mathbf{P}\}. \quad (2.5)$$

The symbol x represents a “generic element” of X and the vertical line $|$ is short for the term “such that.” For example, let the universal set be the set \mathbb{N} of all natural numbers. Then

$$O^+ = \{x \mid x \in \mathbb{N} \text{ and } x = 2y + 1 \text{ for some } y \in \mathbb{N}\} \quad (2.6)$$

defines the set of all odd nonnegative numbers $\{1, 3, 5, \dots\}$. In short,

$$O^+ = \{x \mid x = 2y + 1 \text{ for some } y \in \mathbb{N}\}. \quad (2.7)$$

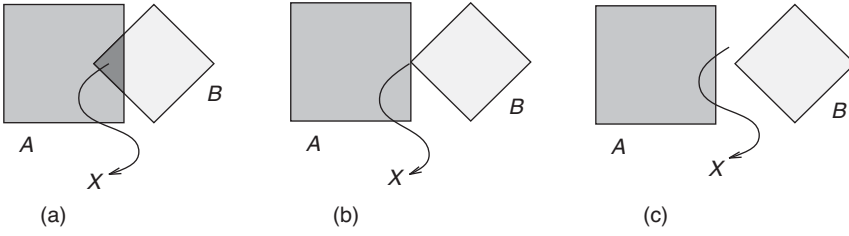


Figure 2.1 (a) X contains many points; (b) X is a singleton; (c) X is an empty set

3. Given a set I , we say that I serves as an *index set* for the family $\mathcal{F} = \{A_\alpha\}$ of sets if, for every $\alpha \in I$, there exists a set of A_α in the family \mathcal{F} . The index set I can be any set, finite or infinite. Very often, we use the set of nonnegative integers $\{1, 2, 3, \dots\}$ as an index set.

The *cardinality* of set A , written $|A|$, is the number of elements in A . Note that the definition of cardinality of a set is unambiguous for a finite set, but that it leads to some problems for sets with an infinite number of elements.

2.1.4 Special Sets

We may meet a set that contains a single element – say, $x \in S$. Such a set is called a *singleton* and is denoted by $\{x\}$. The singleton set $\{x\}$ must not be confused with the element x of S .

It is logically possible (and, in fact, required) to define a set that contains no elements whatsoever. This set is called the *empty set* or *null set*, and is denoted by the symbol \emptyset .

Consider, for example, the case in which two plane regions A and B in the Euclidean plane are given and a set X is defined as follows:

$$X = \{x \mid x \in A \text{ and } x \in B\}. \quad (2.8)$$

Depending on A and B , the set X may have many points, just one point, or no points at all (Figure 2.1).

Note: Given a universal set S , it is perfectly legitimate to define sets whose elements are themselves sets. For example, let S be the Euclidean plane. We define X to be the set of all equilateral triangles. Note that each equilateral triangle is itself a set; namely, a well-defined set of points of S . However, when we talk about X , we do not consider S . Instead, we tacitly abandon our universal set S and implicitly assume a new universal set S_1 that consists of all the triangles on the plane.

2.2 Equality and Inclusion of Sets

Definition 2.1: Two sets A and B are *equal* if and only if every element of A is an element of B and every element of B is an element of A . We write $A = B$. \square

For example, $\{1, 2, 3\} = \{3, 2, 1\}$, but $\{1, 2, 3\} \neq \{1, 2\}$. In particular, note that the order in which the elements appear between braces is unimportant.

This is also a good opportunity to mention that implicit in the definition of set equality is the notion that repetition of elements in a set has no significance. Therefore, if $A = \{1, 1, 2, 2, 3\}$ and $B = \{1, 2, 3\}$, then it is still the case that $A = B$.

(For some applications such as sorting, it is necessary to deal with collections in which the multiplicity of elements does have significance. Such collections have been called *multisets* and *bags* in the literature. In case of multisets, $\{1, 1, 2, 2, 3\} \neq \{1, 2, 3\}$. In this book, however, we do not deal with multisets at any time.)

Definition 2.2: A set A is said to a *subset* of a set B if and only if $x \in A$ implies that $x \in B$; that is, every element of A is an element of B . This can also be expressed as “ A is included in B ” or “ B includes A .” Symbolically, this relation is denoted by $A \subseteq B$ or, equivalently, by $B \supseteq A$. If $A \subseteq B$ and $A \neq B$, A is said to be a *proper subset* of B and is denoted by $A \subset B$. \square

The following are some of the important properties of set inclusion and equality:

$X \subseteq S$ and $\emptyset \subseteq X$, where X is a given set, S universal set, \emptyset empty set;

$A \subseteq A$, where A is any set;

If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$, where A, B, C are any sets.

Let us give a few examples of sets and subsets that we encounter frequently in the context of shape description.

Example 2.1. The set of all points of our perception is generally called *three-dimensional geometric space* and is denoted by \mathbf{R}^3 . Similarly, the set \mathbf{R}^2 of all points on a plane (generally called a *plane*) and the set \mathbf{R}^1 of all points on a line (generally called a *line*) are also examples of sets. Note that

$$\mathbf{R}^2 \subset \mathbf{R}^3, \quad \mathbf{R}^1 \subset \mathbf{R}^3, \quad (2.9)$$

but

$$\mathbf{R}^1 \subset \mathbf{R}^2, \quad \text{if and only if } \mathbf{R}^1 \text{ lies on the plane } \mathbf{R}^2. \quad (2.10)$$

In a similar spirit, the set of all points on a line \mathbf{R}^1 between two points p, q on this line (including p and q) is called the *line segment* between p and q and is denoted by \overline{pq} , or simply pq . Clearly, $\overline{pq} \subseteq \mathbf{R}^1$.

If $p = q$, then the line segment degenerates into a single point. \square

Example 2.2. The set $[a, b]$ of all real numbers greater than or equal to a given number a and less than or equal to a given number b is called the *closed interval* between a and b . Obviously, $[a, b] \subseteq \mathbb{R}$, where \mathbb{R} denotes the set of all real numbers. \square

Example 2.3. We can envisage the set F of all algebraic equations of the form $f(x) = 0$. Similarly, we can consider the set F_a of all algebraic equations $f(x) = 0$ that have a given number a as one solution. $F_a \subset F$. \square

Example 2.4. Throughout the text, we make frequent reference to several well-known sets, which are defined as follows:

$$\mathbb{N} = \{0, 1, 2, 3, \dots\} = \{\text{the natural numbers}\},$$

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \dots\} = \{\text{the integers}\},$$

$$\mathbb{N}^+ = \{1, 2, 3, \dots\} = \{\text{the positive integers}\},$$

$$\mathbb{R} = \{\text{the real numbers}\},$$

$$\mathbb{C} = \{\text{the complex numbers}\}$$

$$\mathbb{Q} = \{\text{the rational numbers}\} = \{x \mid x = \frac{a}{b} \text{ and } a \in \mathbb{Z}, b \in \mathbb{N}^+\},$$

$$\mathbb{Z}_n = \{\text{the integers modulo } n\} = \{0, 1, 2, \dots, (n-1), \text{ where } n \geq 2\}.$$

Note that $\mathbb{Z}_n \subset \mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$.

In our subsequent discussions, we shall frequently use the set of all *positive* real numbers \mathbb{R}^+ and the set of all *negative* real numbers \mathbb{R}^- ; a real number x is called positive if $x > 0$ and negative if $x < 0$. \square

2.3 Some Operations on Sets

In this section, we describe how we can construct new sets from some given sets. To do this, we have to define some operations on sets that will take the given sets as input and will produce a new set as output. The following are some of the more frequently used operations.

2.3.1 The Power Set

Given any set A , we know that the null set \emptyset and the set A are both subsets of A . Also, for any element $a \in A$, the set $\{a\}$ is a subset of A . Quite often, not only one or a few, but all the subsets of a given set, are needed for some investigation.

Definition 2.3: The set of all subsets of a given set A is called the *power set* of A and is denoted by $\mathcal{P}(A)$. Thus

$$\mathcal{P}(A) = \{X \mid X \subseteq A\}. \quad (2.11)$$

\square

For example,

$$\mathcal{P}(\{a, b, c\}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$$

Sometimes, the power set $\mathcal{P}(A)$ is also denoted by 2^A .

The concept of the power set is very important in the context of shape description. Any geometric shape that we see around us is an element of $\mathcal{P}(\mathbf{R}^3)$, where \mathbf{R}^3 denotes three-dimensional geometric space. Similarly, any two-dimensional shape in a plane is an element of $\mathcal{P}(\mathbf{R}^2)$, and any line segment on a line is an element of $\mathcal{P}(\mathbf{R}^1)$.

The power set is an *unary* operation.

2.3.2 Set Union

Definition 2.4: For any two given sets A and B , the *union* of A and B , written as $A \cup B$, is the set of all elements that are members of the set A or the set B or both. Symbolically,

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}. \quad (2.12)$$

□

For example, if $A = \{1, 2, 3, 4, 5\}$ and $B = \{0, 3, 5, 8\}$, then $A \cup B = \{0, 1, 2, 3, 4, 5, 8\}$.

2.3.3 Set Intersection

Definition 2.5: The *intersection* of any two given sets A and B , written as $A \cap B$, is the set consisting of all the elements that belong to both A and B . Symbolically,

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}. \quad (2.13)$$

□

For example, if $A = \{1, 2, 3, 4, 5\}$ and $B = \{0, 3, 5, 8\}$, then $A \cap B = \{3, 5\}$.

If A and B have no common elements – that is, $A \cap B = \emptyset$ – then we say that A and B are *disjoint*.

We can now state two fundamental properties of the power set:

$$\mathcal{P}(A) \cap \mathcal{P}(B) = \mathcal{P}(A \cap B), \quad (2.14)$$

$$\mathcal{P}(A) \cup \mathcal{P}(B) \subseteq \mathcal{P}(A \cup B). \quad (2.15)$$

2.3.4 Set Difference

Definition 2.6: Let A and B be any two given sets. The set *difference* of A and B , written as $A - B$, is the set consisting of all elements of A that are not elements of B . Thus

$$A - B = \{x \mid x \in A \text{ and } x \notin B\}. \quad (2.16)$$

□

For example, if $A = \{1, 2, 3, 4, 5\}$ and $B = \{0, 3, 5, 8\}$, then $A - B = \{1, 2, 4\}$.

2.3.5 Set Complement

The notion of set complement could be derived from the idea of set difference.

Definition 2.7: Let A be any given set. The set of all elements x that belong to the universal set S but do not belong to A is called the *complement* of A and is denoted by A^c . Symbolically,

$$A^c = \{x \mid x \in S \text{ and } x \notin A\}. \quad (2.17)$$

□

Note that

$$A^c = S - A. \quad (2.18)$$

This is why $A - B$ is also called the *relative complement* of B in A (or of B with respect to A), provided that B is a subset of A .

2.3.6 Symmetric Difference

Definition 2.8: Let A and B be any two given sets. The *symmetric difference* of A and B is defined as

$$A \Delta B = (A \cup B) - (A \cap B). \quad (2.19)$$

□

$A \Delta B$ is the set of all elements that are members of exactly one of the sets A and B . The symmetric difference is also called the *Boolean sum* of A and B , because

$$A \Delta B = (A - B) \cup (B - A). \quad (2.20)$$

The symmetric difference has some interesting properties:

$$A \Delta B = B \Delta A, \quad (A \Delta B) \Delta C = A \Delta (B \Delta C), \quad A \Delta \emptyset = A, \quad A \Delta A = \emptyset.$$

Note: Even though the symmetric difference operator has such nice properties, up to the present time it has hardly been used as a shape operator, whereas we see frequent use of other set operators as shape operators. One reason for this is certainly its composite nature, which makes it less intuitive than the others.

2.3.7 Venn Diagrams

Some of the above operations can be depicted pictorially with the help of *Venn diagrams* (Figure 2.2). A Venn diagram is a schematic representation of a set by a set of points. The universal set S is generally represented by a set of points in a rectangle, while a subset A of S is generally represented by the interior of a circle or some simple shape.

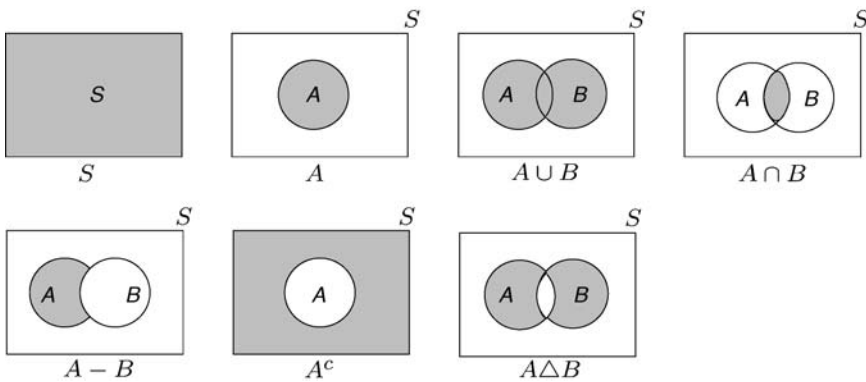


Figure 2.2 Venn diagrams of a few constructed shapes

We list below a few fundamental properties of some of the set operations:

Union : $A \cup B = B \cup A$ (commutative), $A \cup \emptyset = A$, $A \cup A = A$,
 $(A \cup B) \cup C = A \cup (B \cup C)$ (associative);

Intersection : $A \cap B = B \cap A$ (commutative), $A \cap \emptyset = \emptyset$, $A \cap A = A$,
 $(A \cap B) \cap C = A \cap (B \cap C)$ (associative);

Complement : $(A^c)^c = A$, $\emptyset^c = S$, $S^c = \emptyset$,
 $A \subseteq B$ implies $B^c \subseteq A^c$;

Difference : $A = (A \cap B) \cup (A - B)$, $B \cap (A - B) = \emptyset$;

Combinations : $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$,
 $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$, $A \cup A^c = S$, $A \cap A^c = \emptyset$,
 $(A \cup B)^c = A^c \cap B^c$ $(A \cap B)^c = A^c \cup B^c$.

You should have no difficulty in verifying these rules, either from the definitions or with the help of Venn diagrams.

2.3.8 Cartesian Products

Apart from the set operations of union, intersection, and so on, another, rather different, and very important method of constructing new sets from given sets exists. To discuss this method, we first need the concept of an *ordered pair*. An ordered pair (a, b) consists of two objects a and b , written in this order. The ordered pairs (a, b) and (x, y) are said to be equal if and only if $a = x$ and $b = y$. From this, it follows that $(a, b) \neq (b, a)$ unless the objects a and b are identical.

The idea of an ordered pair can be extended to define an ordered triple, and, more generally, an n -tuple. An ordered n -tuple (a_1, a_2, \dots, a_n) consists of n objects a_1, a_2, \dots, a_n , written in this order. Alternatively, an ordered n -tuple can be defined to be an ordered pair whose first member is an ordered $(n - 1)$ -tuple; that is, as $((n - 1)\text{-tuple}, a_n)$.

Definition 2.9: Let A and B be two arbitrary (nonempty) sets. We define the *Cartesian product* $A \times B$ of these two sets as the collection of all ordered pairs (a, b) that can be formed by taking a first term from A and a second term from B . Thus,

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}.$$

□

For example, if $A = \{\alpha, \beta\}$ and $B = \{1, 2, 3\}$, then

$$A \times B = \{(\alpha, 1), (\alpha, 2), (\alpha, 3), (\beta, 1), (\beta, 2), (\beta, 3)\}. \quad (2.21)$$

In general, $A \times B$ is “bigger” than A or B , because if A has n elements and B has m elements, then $A \times B$ has $n \cdot m$ elements. The following results can also be noted:

$$A \times (B \cup C) = (A \times B) \cup (A \times C), \quad (2.22)$$

$$A \times (B \cap C) = (A \times B) \cap (A \times C), \quad (2.23)$$

$$A \times (B - C) = (A \times B) - (A \times C). \quad (2.24)$$

Note: As we all know, the notion of the Cartesian product plays a very crucial role in the study of geometry. Consider the following example. Let A be the closed interval $[a, b]$ – that is, the set of all real numbers x such that $a \leq x \leq b$ – and let B be the closed interval $[c, d]$ – that is, the set of real numbers y such that $c \leq y \leq d$. In the manner of Cartesian analytic geometry, we can model these sets as portions of two straight lines on a plane, say the “ x -axis” and the “ y -axis,” respectively. (We have not yet explained whether or not this is a valid model; the validity of the model will be established later.) The product set $A \times B$ can then be thought of as a portion of the xy -plane (Figure 2.3(a)). It is not necessary, in fact, to consider the x -axis and the y -axis to be perpendicular to each other. We can equally well imagine any two lines on the plane to represent A , B , and $A \times B$ (Figure 2.3(b)).

It is certainly possible to take the Cartesian product $A \times A$ of a set A with itself. For example, the Cartesian product of the set \mathbb{R} of all real numbers with itself is the set of all ordered pairs (x, y) of real numbers. We usually denote $\mathbb{R} \times \mathbb{R}$ by \mathbb{R}^2 and frequently refer to it as the *Cartesian coordinate plane*.

Note: It is perhaps necessary to comment on one possible source of misunderstanding. Because of our previous experience in analytic geometry, we always consider a point p in the geometric plane \mathbf{R}^2 (which is a geometric object) and a corresponding ordered pair (x, y) of real numbers in the Cartesian coordinate plane \mathbb{R}^2 (which is an algebraic object) as being – to all intents and purposes – identical to one another. But \mathbb{R}^2 is a pure set and has no structure whatever, since no structure has yet been assigned to it. On the other hand, the geometric plane \mathbf{R}^2 is a set that has some strong algebraic and analytical structures. To use \mathbb{R}^2 as a valid model of \mathbf{R}^2 , it is therefore necessary to add some structures to the former set. For example, we frequently add a measure structure to it – in other words, we introduce the notion of *distance* between any two ordered pairs (x_1, y_1) and (x_2, y_2) – which endows the set \mathbb{R}^2 with a certain “spatial” character. Remember that the distance between two ordered pairs can be defined in various different ways. If the distance is defined to be equal to $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, the resulting set is called the *Euclidean plane* instead of the Cartesian coordinate plane.

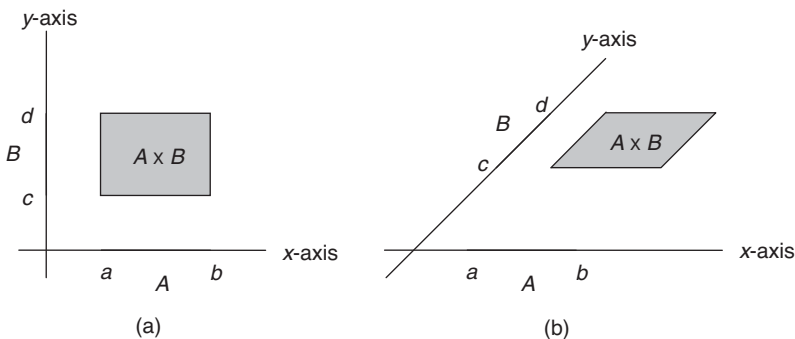


Figure 2.3 The modeling of the Cartesian product $A \times B$ of two closed intervals $[a, b]$ and $[c, d]$

There is another set that is closely connected with the Cartesian coordinate plane \mathbb{R}^2 . This is the *complex plane*. If z is a complex number of the form $x + iy$, where x and y are real numbers, then we can identify z with the ordered pair (x, y) , and thus with an element of \mathbb{R}^2 . As a whole, the set \mathbb{C} of all complex numbers can be identified with the Cartesian coordinate plane \mathbb{R}^2 . But the complex numbers are more than merely a set. They constitute a number system, with operations of addition, multiplication, conjugation, and so on; that is, they have a richer algebraic structure. When the coordinate plane \mathbb{R}^2 is thought of as consisting of complex numbers and is enriched by the algebraic structure that it acquires from complex numbers, it is called the *complex plane*. Traditionally, the letter \mathbb{C} is used to denote either the set of complex numbers or the complex plane.

2.4 Relations in Sets

2.4.1 Fundamental Concepts

A subset R of $A \times B$ is called a *relation* between the elements of a set A and those of another set B .

Consider, for example, two sets $A = \{1, 2\}$ and $B = \{1, 2, 3\}$. Then $A \times B = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3)\}$. Let us now take the following subset R of $A \times B$: $R = \{(1, 2), (1, 3), (2, 3)\}$. We call R a relation between the elements of A and B .

You might wonder why the term “relation” is being used in such a situation, since it does not appear to match with our everyday concept of a relation. But indeed it does match.

Examine the subset R in the above example again. For all pairs $(a, b) \in R$, the following proposition is true: “ a is smaller than b .” And these are the only elements in $A \times B$ for which the above proposition holds. Thus R expresses a relation (in this particular example, the relation “smaller than”) between the elements of A and B .

Most of our familiar examples of relations could be given a name. For example, “greater than,” “left of,” “top of,” “father of,” “divided by,” “being married to,” and so on. But explicit naming of a relation is an undue restriction. The above example clearly shows that a relationship among the elements of two sets can be established by simply writing down a list of all those pairs that we want to call related.

In mathematical terminology, if R is any subset of $A \times B$, and if $(a, b) \in R$, we say that “ a is in relation R to b ,” or simply “ aRb .”

The above examples and the related text suggest relationships between two objects. That is why such relations are also called *binary relations*. The relation between two parents and a child, the coincidence of three lines, and that of a point lying between two points are examples of relations between three objects. In fact, we can generalize the concept of relation as follows:

Definition 2.10: An n -ary relation R on the sets A_1 to A_n is a subset of the Cartesian product $A_1 \times A_2 \times A_3 \times \cdots \times A_n$.

In other words, the objects x_1, x_2, \dots, x_n (where $x_1 \in A_1, x_2 \in A_2, \dots$) are related by R if and only if the ordered n -tuple $(x_1, x_2, \dots, x_n) \in R$. \square

For a binary relation R , the two sets A and B need not be distinct. These two sets may be the same set (say, $= A$). Then we say that R is a subset of $A \times A$ or A^2 , which defines a “relation of A .”

For example, let \mathbb{Z} be the set of integers and let $R \subseteq \mathbb{Z} \times \mathbb{Z}$ be given by

$$R = \{(a, b) \mid a \neq 0, b = na \text{ for some integer } n\}.$$

Here, the relation aRb means “ a divides b .”

Note that, in general, the number of different relations on a set A depends on the cardinality of A ; that is, on $|A|$. Most of these relations will not be of any particular interest. However, for a given set we can always derive three special relations:

$R = A \times A$, called the *universal relation* U_A on A ;

$R = \emptyset$, called the *empty relation* on A ;

$R = \{(a, a) \mid a \in A\}$, called the *identity relation* I_A on A .

2.4.2 The Properties of Binary Relations in a Set

Obviously, general relations, being just a subset of a product of sets, are not particularly interesting, as very little can be said about them. However, when the relations satisfy further conditions, they become more interesting. Let us consider a few fundamental properties with which relations may be endowed. Each property is said to be present whenever the corresponding condition is satisfied.

Definition 2.11: A relation R on a set A is *reflexive* if aRa for every $a \in A$; that is, if every element of A stands in relation R to itself. This implies that $(a, a) \in R$. \square

Example 2.5. Let A be the set of all real numbers and let the relation be “less than or equal to” (represented by the symbol “ \leq ”). In this case, for any element a , $a \leq a$. Therefore, this relation is reflexive. But if we choose the relation “ $<$,” it is not reflexive. \square

Example 2.6. Let A be the set of all triangles in a plane. The relation “triangle a is congruent to triangle b ” is a reflexive relation. Also, the relation “triangle a is similar to triangle b ” is a reflexive one. \square

Example 2.7. Let A be the set of all lines in a plane and let the relation be “is parallel to” (denoted symbolically by “ \parallel ”). If we consider any line as being parallel to itself, then it is a reflexive relation.

On the other hand, if the relation is “is perpendicular to” (i.e., \perp), then it is obviously not a reflexive relation. \square

Definition 2.12: A relation R on a set A is *symmetric* if, for every $a, b \in A$, whenever aRb , then bRa . In other words, aRb implies bRa . \square

Example 2.8. The relations \leq and $<$ are not symmetric on the set of real numbers, while the relation of equality (i.e., “ $=$ ”) is. \square

Example 2.9. The relations “is congruent to” and “is similar to” on the set of all triangles are also symmetric. \square

Example 2.10. The relation \parallel is symmetric. So too is the relation \perp , since if $a \perp b$, then certainly $b \perp a$. \square

Definition 2.13: A relation R on a set A is *transitive* if, for every $a, b, c \in A$, whenever aRb and bRc , then aRc . \square

Example 2.11. The relations \leq , $<$, and $=$ are transitive on the set of real numbers. \square

Example 2.12. The relations “is congruent to” and “is similar to” on triangles are both transitive. \square

Example 2.13. The relation \perp is not transitive, for if $a \perp b$ and $b \perp c$, then $a \parallel c$ rather than $a \perp c$. The relation \parallel on lines is, however, transitive. \square

Definition 2.14: A relation R on a set A is *antisymmetric* if, for every $a, b \in A$, whenever aRb and bRa , then $a = b$. In other words, aRb and bRa implies $a = b$. \square

Example 2.14. The relation \leq on the set of real numbers is antisymmetric. \square

Example 2.15. Consider the following relation:

$$R = \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a \text{ divides } b\}.$$

This relation is antisymmetric. \square

Example 2.16. Let X be a set of furniture in a drawing room and let L be a relation as given below:

$$L = \{(x_i, x_j) \mid x_i, x_j \in X \text{ and } x_i \text{ is placed at the left of } x_j\}.$$

If we refer to Figure 2.4(a), where the objects are arranged linearly, it may appear that the relation L is both antisymmetric and transitive. On the other hand, if the objects are arranged circularly (Figure 2.4(b)), and if we assume that L is transitive, then $x_1 L x_6$ as well as $x_6 L x_1$. This means that the relation is not antisymmetric. The problem arises here because our intuitive notion of the “left of” relation is ambiguous. \square

Note: The properties “symmetry” and “antisymmetry” are not mutually exclusive. It is possible to have a relation that is both symmetric and antisymmetric. For example, for any set A , the identity relation I_A is both symmetric and antisymmetric.

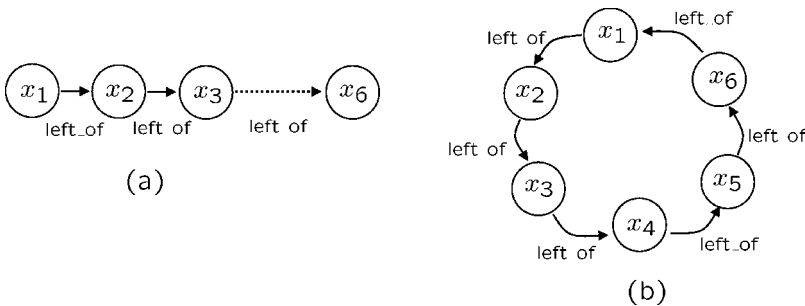


Figure 2.4 The spatial arrangement of furniture in a drawing room

2.4.3 Equivalence Relations and Partitions

The most important type of relation that is frequently encountered in all branches of pure and applied mathematics, particularly in geometry, is called an equivalence relation.

Definition 2.15: A relation on a set is said to be an *equivalence* relation if and only if it is reflective, symmetric, and transitive. \square

It is customary to use the special symbol “ \sim ” to denote an equivalence relation (in place of the general symbol R). Thus, if $a, b, c, \dots \in A$ and \sim is an equivalence relation on A , then:

$$(i) \ a \sim a \text{ for all } a \in A \text{ (reflective);} \quad (2.25)$$

$$(ii) \ a \sim b \text{ implies } b \sim a \text{ (symmetric);} \quad (2.26)$$

$$(iii) \ a \sim b \text{ and } b \sim c \text{ imply } a \sim c \text{ (transitive).} \quad (2.27)$$

(In some of the literature, the letter E is also used to denote an equivalence relation. Thus we write aEb instead of $a \sim b$.)

Example 2.17. The relations “is congruent to” and “is similar to” on the set of all triangles in the plane are both equivalence relations. \square

Example 2.18. The $=$ (“equal to”) relation on the set of all real numbers is certainly an equivalence relation. \square

Example 2.19. Consider the set A of all points in a vertical plane. The relation “point a is at the same height as the point b ” is an equivalence relation. \square

One reason for the importance of equivalence relations is that they permit the separation of the set into convenient subsets. In order to clarify this point, we need the following concepts.

Definition 2.16: Let X be a given set and $A = \{A_1, A_2, \dots, A_m\}$ where each A_i , ($i = 1, \dots, m$) is a subset of X . We say that the set A is a *cover* of the set X if

$$\bigcup_{i=1}^m A_i = X. \quad (2.28)$$

The notation $\bigcup_{i=1}^m A_i$ denotes $A_1 \cup A_2 \dots \cup A_m$. \square

If, in addition, the elements of A , which are subsets of X , are mutually disjoint (i.e., $A_i \cap A_j = \emptyset$ if $i \neq j$), then A is called a *partition* of X , and the sets A_1, A_2, \dots, A_m are called the *blocks* of the partition.

For example, let $X = \{a, b, c\}$ and consider the following collections of subsets of X :

$$A = \{\{a, b\}, \{b, c\}\}, \quad B = \{\{a\}, \{a, c\}\}, \quad C = \{\{a\}, \{b, c\}\}.$$

The sets A and C are covers of X , while B is not. Only the set C is a partition of X , which has two blocks in it. Of course, every partition is also a cover.

Consider one more example. A partition for the set of all integers \mathbb{Z} is given by $\{\mathbb{N}^+, \mathbb{N}^-, \{0\}\}$, where \mathbb{N}^+ and \mathbb{N}^- are the sets of positive and negative integers. Another partition of \mathbb{Z} would be $\{E, O\}$; that is, into even numbers E and odd numbers O . We can, of course, imagine several partitions of a set.

We are now in a position to state the crucial fact about equivalence relations:

Theorem 2.1. *Any equivalence relation on a set A leads to a unique partition of A . Conversely, any given partition of a set A defines an equivalence relation on A .*

Let us clarify the idea of the theorem by means of a few examples.

Example 2.20. Let L be the set of all lines in a plane and let the relation be \parallel . We know that it is an equivalence relation. For every $a \in L$, we can form the set of all lines in the plane parallel to the line a . Let the set be denoted by L_a . In this way, the whole set L can be partitioned into sets of parallel lines L_a, L_b, L_c, \dots , as indicated in Figure 2.5. \square

Example 2.21. Let V be the set of all points in a vertical plane and let the relation be “is at the same height.” We know that this is an equivalence relation. This particular equivalence relation partitions the vertical plane V into horizontal lines. In other words, the blocks of the partition are horizontal lines. \square

Example 2.22. Let P be the set of all points in a plane. Two points a and b are defined as related if they are equidistant from some point, say o , in the plane. This is an equivalence relation that partitions the plane into concentric circles. \square

The notion of the *equivalence class* naturally emerges at this point.

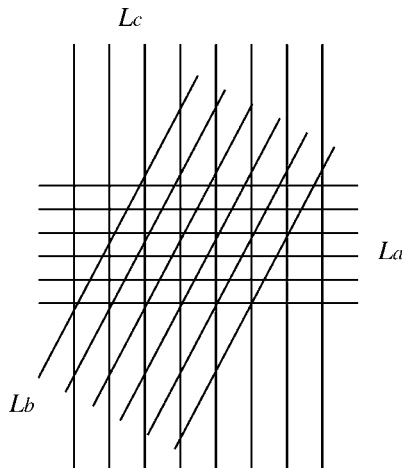


Figure 2.5 Partitioning all the lines in a plane into sets of parallel lines

Definition 2.17: Let E be an equivalence relation on a set A . Let x be a fixed (but arbitrary) element of A . Consider the collection of all elements of $y \in A$ that are related to the given x . This collection (a subset of A) is called the *equivalence class* of x (or the E -equivalence class generated by x). A standard notation for this class is $[x]$, so that

$$[x] = \{y \mid y \in A \text{ and } yEx\}. \quad (2.29)$$

□

The particular x that generates the equivalence class $[x]$ is often referred to as a *representative element* of the class.

We know that every equivalence relation E partitions the set A into a number of disjoint subsets or blocks. It is not difficult to see that the blocks of the partition correspond to the E -equivalence classes. Mathematicians use a special name, the *quotient set*, to denote these blocks as a collection.

Definition 2.18: The collection of all (distinct) equivalence classes induced on A by E is called the *quotient set* of A modulo E and is denoted by A/E . Thus,

$$A/E = \{\dots, [x], \dots\}. \quad (2.30)$$

A/E is simply the set of blocks of the partition. □

The fact that “there is no essential distinction between a partition and an equivalence relation – either one determines the other” has a profound influence on the study of geometry, and particularly on shape description and analysis. We shall come back to this point later, in the next section.

2.4.4 Order Relations

Just as the notion of “equality” (between, say, numbers) gives rise to the mathematical concept of equivalence, an *order relation* is a generalization of the notion of “inequality” on numbers.

Definition 2.19: A binary relation R on a set A is called a *partial order relation* if R is reflexive, antisymmetric, and transitive. □

It is customary to use the special symbol \leq for partial order relations. Thus, if $a, b, c \in A$ and \leq is an partial order relation on A , then:

- (i) $a \leq a$ for all $a \in A$ (*reflexive*);
- (ii) $a \leq b$ and $b \leq a$ imply $a = b$ (*antisymmetric*);
- (iii) $a \leq b$ and $b \leq c$ imply $a \leq c$ (*transitive*).

Note that the symbol \leq for partial ordering does not necessarily mean “less than or equal to,” although the most familiar partial order relation is precisely the “less than or equal to” relation on the real numbers. Another point should also be noted. It is sometimes convenient to write $b \geq a$, instead of $a \leq b$; both mean precisely the same.

If $a \leq b$ and $a \neq b$, we usually write $a < b$. This relation on the real numbers is usually read as “strictly less than.”

Definition 2.20: A set A with a partially order relation \leq defined on it is called a *partially ordered set*, or simply a *poset*. If, in addition, for every pair $a, b \in A$, we have either $a \leq b$ or $b \leq a$, we call A a *totally ordered set* or a *chain*. \square

We will now present a few partial order relations that are frequently used in practice.

Example 2.23. Let R be the set of real numbers. The relation “less than or equal to” is a partial ordering on R . In fact, it is a total ordering or chain. \square

Example 2.24. Let \mathbb{N}^+ be the set of positive integers. Define “ $a \leq b$ if and only if a divides b .” This order relation makes \mathbb{N}^+ a poset, but not a chain, because for every pair a and b , we cannot have $a \leq b$ or $b \leq a$. For example, $a = 5$ and $b = 7$ are incomparable. \square

Example 2.25. Another important partial order relation in geometry is the *set inclusion* relation (\subseteq). Let $\mathcal{P}(X)$ be the power set of an arbitrary set X . Then set inclusion $A \subseteq B$ on $\mathcal{P}(X)$ is a partial order relation. But the order is not total, since for two given sets A, B we need not have either $A \subseteq B$ or $B \subseteq A$ (Figure 2.6). Thus, $\mathcal{P}(X)$ is a poset, but not a chain. \square

We will now discuss the very important notion of *bounds*, which arises from the concept of partial order relations.

Definition 2.21: Let X be a partially ordered set and let $A \subseteq X$. Any element $x \in X$ is an *upper bound* for A if for all $a \in A$, $a \leq x$. Similarly, any element $x \in X$ is a *lower bound* for A if, for all $a \in A$, $x \leq a$. \square

Example 2.26. Let $L = \{a, b, c\}$ be some set and X be the power set of L ; that is,

$$X = \mathcal{P}(L) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, L\}.$$

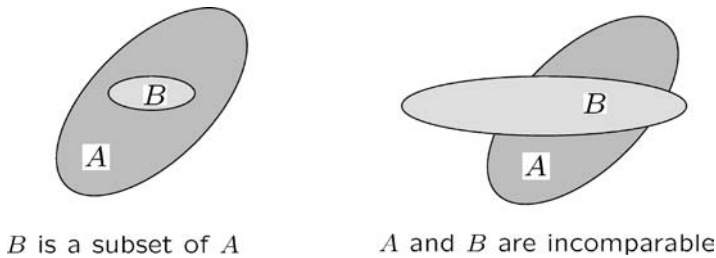


Figure 2.6 $\mathcal{P}(X)$ is a poset, not a chain

Let a subset A of X be given by

$$A = \{\{b\}, \{c\}, \{b, c\}\}.$$

Then the upper bounds for A are $\{b, c\}$ and L , while \emptyset is the only lower bound.

For the subset $B = \{\{c\}, \{a, c\}\}$, the upper bounds are $\{a, c\}$ and L , while the lower bounds are \emptyset and $\{c\}$. \square

Example 2.27. Let $X = \{4, 6, 12, 24, 48, 72\}$ and let the partial order relation be “ $a \leq b$ if and only if a divides b .” Let a subset A of X be given by

$$A = \{4, 6, 12\}.$$

Then 12, 24, 48, 72 are upper bounds for A , and there is no lower bound. \square

Note that the upper and lower bounds of a subset may not exist at all. Moreover, the upper and lower bounds are not necessarily unique – there may be several upper and lower bounds of a subset. Therefore, we define the following terms.

Definition 2.22: Let X be a partially ordered set and let $A \subseteq X$. An element $x \in X$ is a *least upper bound* or *supremum* for A if x is an upper bound for A and $x \leq y$, where y is any upper bound for A .

Similarly, the *greatest lower bound* or *infimum* for A is an element $x \in X$ such that x is a lower bound and $y \leq x$ for all lower bounds y . \square

The standard notations for these terms are $\sup(A)$ and $\inf(A)$, respectively.

If $A \subseteq X$ has any \sup (\inf), it is said to be bounded from above (below). If both exist, we simply say that “ A is bounded.”

Example 2.28. For a totally ordered set or a chain, every subset has a supremum and an infimum. \square

Example 2.29. Let \mathbb{Q} be the set of all rationals and let $A \subseteq \mathbb{Q}$, with elements q defined by $2 < q^2 < 3$. There are infinitely many upper and lower bounds (namely, rationals $r \in \mathbb{Q}$ such that $r > \sqrt{3}$ or $r < \sqrt{2}$). But these sets have no supremum or infimum. To understand the last statement, note that if we say that a rational number r_1 is the supremum, then there always exists another rational number r_2 that is an upper bound (i.e., $r_2 > \sqrt{3}$) and $r_2 \leq r_1$. This means that r_1 cannot be the supremum. Similar reasoning also holds for the infimum. The basic problem here is that $\sqrt{3}$ and $\sqrt{2}$ are not rational numbers.

Now consider the following example. Let \mathbb{R} be the set of real numbers and let $A \subseteq \mathbb{R}$ be the set of rational numbers x such that $2 < x^2 < 3$. Then there exist the $\sup(A) = \sqrt{3}$ and the $\inf(A) = \sqrt{2}$. Note that $\sup(A)$ and $\inf(A)$ do not belong to A . \square

2.5 Functions, Mappings, and Operations

2.5.1 Fundamental Concepts

In this section, we study a particular class of relations called *functions*. It is no exaggeration to say that one of the most basic notions occurring in mathematics is that of function.

Definition 2.23: Let X and Y be any two sets (not necessarily distinct). A relation f from X to Y is called a *function* if, for every $x \in X$, there is a unique $y \in Y$ such that $(x, y) \in f$. \square

Note that the definition of function requires that a relation must satisfy two additional conditions in order to qualify as a function. These conditions are as follows:

- (a) every $x \in X$ must be related to some $y \in Y$;
- (b) each x must be related to one and only one element y .

Therefore, the definition of function can also be restated as follows:

Definition 2.24: A function f is a rule that assigns to each element $x \in X$ one and only one element $y \in Y$. We say that f is a *mapping* from X to Y , and we write

$$f : X \rightarrow Y \quad \text{or sometimes} \quad X \xrightarrow{f} Y. \quad (2.31)$$

\square

Terms such as *mapping*, *operation*, *transformation*, and so on are also used in special cases as synonyms for *function*.

The set X is called the *domain* of the function f , and is often denoted by D_f . This means that $D_f = X$. The set Y is called the *codomain* of f . It is possible that the domain and the codomain may coincide. We then have a function $f : X \rightarrow X$.

For a function $f : X \rightarrow Y$, if $(x, y) \in f$, then x is called the *argument* and the corresponding y is called the *image* of x under f . Instead of writing $(x, y) \in f$, it is customary to write $y = f(x)$ and to call y the *value* of the function f at x .

Note: Although *functional notation* – that is, $f(x)$ – is the most popular way of denoting the image of x under f , several other notations are in use, including the following: *left-operator notation* – that is, fx (e.g., $\sin x$, $\frac{d}{dx}\phi$, $3x$); *right-operator notation* – that is, xf (e.g., $x/2$); *superscript notation* – that is, f^x (e.g., e^x); and *subscript/suffix notation* – that is, f_x (e.g., $\frac{d}{dx}\phi$ is frequently denoted by ϕ_x).

Let us now present a few simple examples of functions.

Example 2.30. Let X be any set; define $f : X \rightarrow X$ by $x = f(x)$. This is called the *identity mapping* of X . Although it seems the most trivial mapping, we shall see later that it is significant in many ways, particularly because it behaves like the number 1 in various situations. Frequently, this function is denoted by the special symbol ι . We can also use I_x to denote this function. \square

Example 2.31. Let \mathbf{R}^2 be the set of all points on a plane. Let l be a line on that plane. Let $f : \mathbf{R}^2 \rightarrow \mathbf{R}^2$ be defined by (where $x \in \mathbf{R}^2$):

$$f(x) = \begin{cases} x, & \text{if } x \text{ is on } l; \\ y, & \text{if } x \text{ is off } l; \end{cases} \quad \text{and } l \text{ is the perpendicular bisector of the line segment } \overline{xy}.$$

Note that this function is simply the *reflection* about a line l on a plane (Figure 2.7). Frequently, this function is denoted by the special symbol σ .

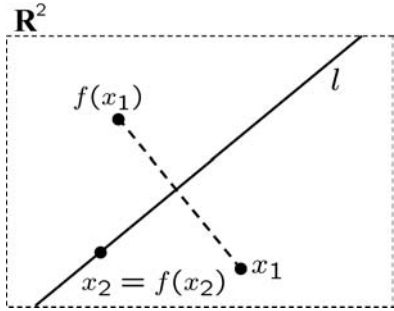


Figure 2.7 Reflection about a line l in \mathbf{R}^2

It is quite obvious that the well-known *geometric transformations*, such as *translation*, *rotation*, *scaling*, and so on, can also be viewed as functions $f : \mathbf{R}^2 \rightarrow \mathbf{R}^2$ (2-D geometric transformations) or $f : \mathbf{R}^3 \rightarrow \mathbf{R}^3$ (3-D geometric transformations), in the same way. □

Example 2.32. Let us consider a spherical zone X and a plane Y (Figure 2.8). Let us define a function $f : X \rightarrow Y$ in the following manner. Assume there is a fixed point – say, the observer’s viewpoint. Connect every point $x \in X$ with the observer’s viewpoint by a straight line. The image of x under f is the intersection of this line with the plane Y . This function is, in fact, a *perspective projection*, which is a function from \mathbf{R}^3 to \mathbf{R}^2 . □

Example 2.33. The previous examples show the extreme generality of the concept of a function. However, not every rule of assignment is a function. Let \mathbb{R} be the set of real numbers and let $X = Y = \mathbb{R}$. Consider the rule “ $y = f(x)$ if and only if $y^2 = x$, where $x \in X$ and $y \in Y$.” This

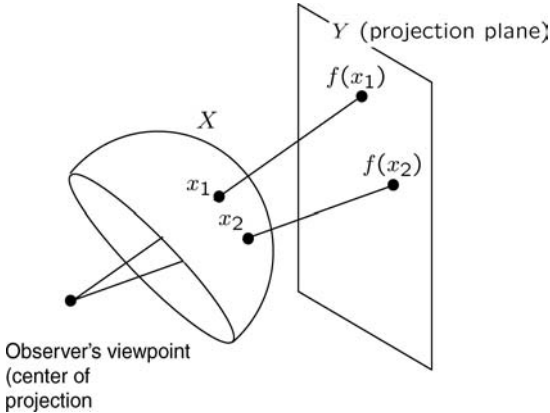


Figure 2.8 The perspective projection from the observer’s viewpoint

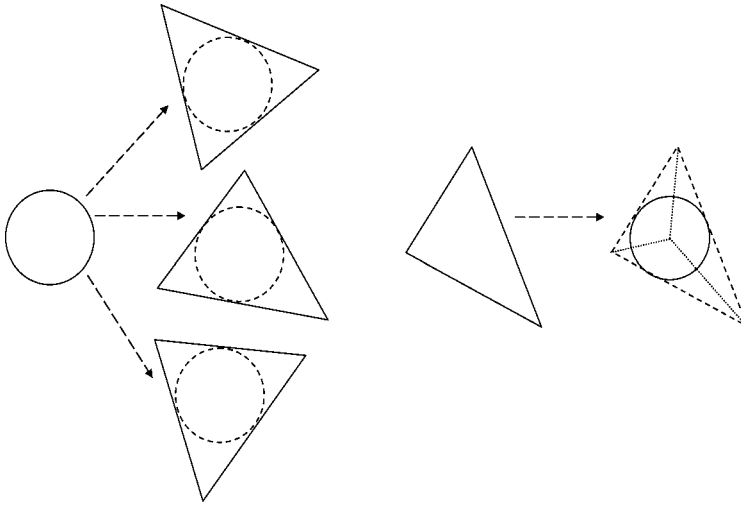


Figure 2.9 Not every rule of assignment is a function: (a) one circle corresponds to many triangles; (b) one triangle corresponds to one circle – while the rule of the assignment in (a) is not a function, that in (b) is a function

is not a function since, for example, it assigns to $x = 9$ two elements of Y ; namely, $+3$ and -3 . The definition of a function demands the *uniqueness* of the assignment.

The same situation may arise in the case of geometric objects too. Let C be the set of all circles and T let be the set of all triangles in a plane. The rule “assign to each circle the corresponding circumscribed triangle” does not define a function, since more than one triangle can be assigned to each circle (Figure 2.9(a)). On the other hand, if the rule is “assign to each triangle the corresponding inscribed circle,” then it defines a function $f : T \rightarrow C$ (Figure 2.9(b)). \square

2.5.2 The Graphical Representations of a Function

Sometimes, it is useful to represent a function by means of a picture. One of the simplest such pictorial representations would be to represent the elements of the set X by points in a left-hand column and the elements of Y by the points in a right-hand column, and then to draw an arrow from a point in the left-hand column to a point in the right-hand column, to indicate that the corresponding element in X is related to the corresponding element in Y . For example, let $X = \{3, 7, G, \square\}$, $Y = \{2, 7, 9, p, \Delta\}$. We can define a function $f : X \rightarrow Y$ as the set $f = \{(3, 2), (7, 9), (G, p), (\square, \Delta)\}$. Its pictorial representation is shown in Figure 2.10(a).

Another popular method of pictorial representation of a function is closely allied to traditional coordinate geometry. Draw a pair of perpendicular axes (commonly the x -axis horizontal and the y -axis vertical) and on each of the axes mark points representing the elements of the sets X and Y , respectively. Now, for each element (x, y) in the function, place a mark on the

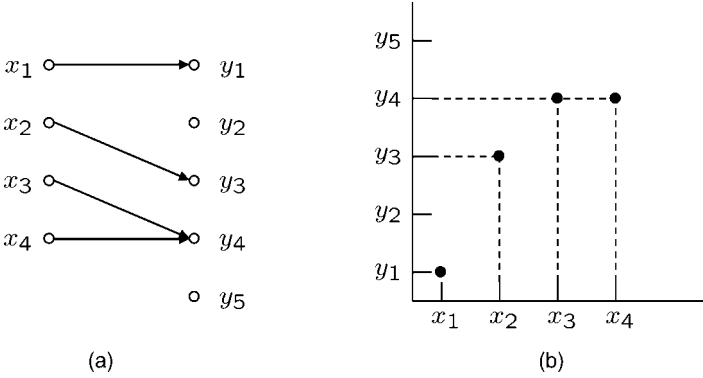


Figure 2.10 Graphs of a function

diagram above the corresponding x on the x -axis and to the right of the corresponding y on the y -axis. Such a representation for the function defined in the previous example is shown in Figure 2.10(b).

It must be noted that such pictorial representations are possible not only for functions but also for any general relation R from X to Y . We call such a representation the *graph* of the function (relation).

The graph of a function, particularly one produced by the second method, is very frequently used to represent a function and to study its characteristics. But we can observe that such a representation may at times give rise to some misconceptions. We will briefly present here a few such commonly occurring misconceptions:

1. Figure 2.11(a) shows the graph of a function $f : A \rightarrow Y$. Although Figure 2.11(b) appears to be the graph of another function, it does not represent a function at all, since there are three y 's corresponding to some single x . We can say that it is the graph of a relation from

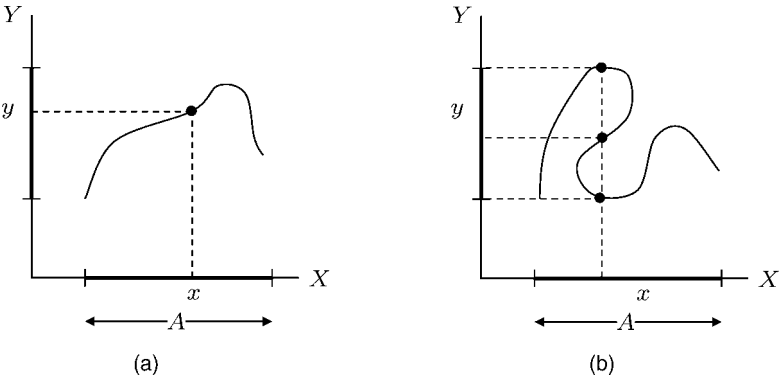


Figure 2.11 (a) The graph represents a function; (b) the graph does not represent a function

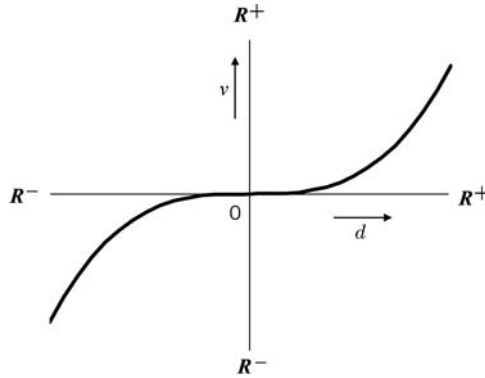


Figure 2.12 The graph of the function $V = \frac{1}{6}\pi d^3$

A to Y , but not the graph of a function. However, usage in the literature has not always been consistent, and at times the phrase *single-valued function* has been used in the sense that we have given for “function,” with references to a *multivalued function* as something in which an element of X can be mapped onto more than one element in Y .

Note: It must be emphasized that the familiar equation of a straight line $y = ax + b$, or that of a parabola $y = ax^2$, are truly functions, while those of a circle ($x^2 + y^2 = a^2$) or an ellipse ($\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$) are not. The latter examples are considered to be multivalued functions.

2. Functions are frequently used to represent the relationships between two real-world objects. For example, consider the question of describing the volume of a sphere in terms of its diameter. Usually, this relationship is defined by a function $f : \mathbb{R} \rightarrow \mathbb{R}$ (where \mathbb{R} denotes the set of all real numbers), such that $V = f(x) = \frac{1}{6}\pi d^3$. Here, V denotes the volume and d denotes the diameter of the sphere. The graph of the function is shown in Figure 2.12.

However, the function f shown above is not an exact description of the real-world situation, since the volume of a sphere cannot be defined for negative values of the diameter. Therefore, if a function is defined for some, but possibly not all, elements of X (i.e., for some arguments in the domain, the function remains undefined), then it is called a *partial function*. On the other hand, if a function is defined for all elements of X (i.e., $D_f = X$), it is said to be a *total function*.

Note that the definition of function $f : X \rightarrow Y$ requires that every $x \in X$ must be related to some $y \in Y$. However, in defining the partial function we are relaxing that requirement. In some cases, given a function $f : X \rightarrow Y$, we may want to construct a new function f' , having as domain a subset X' of X such that

$$f' : X' \rightarrow Y, \quad \text{given by } f'(x') = f(x') \text{ for all } x' \in X' \subset X. \quad (2.32)$$

This f' could be considered as a partial function from X to Y , and is called the *restriction* of f to X' .

(The opposite concept is that of an *extension*. If $f' : X' \rightarrow Y$ and $X \supset X'$, $Z \supset Y$, then any function $f : X \rightarrow Z$ with the property that $f(x') = f'(x')$ for $x' \in X'$ is called an extension of f' from X' to the domain X .)

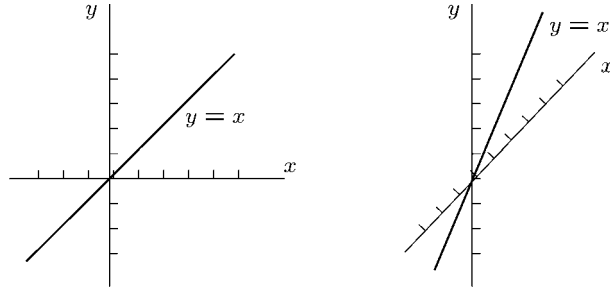


Figure 2.13 Two different graphical representations of the same function $y = x$

In the particular example given above, we can define a total function $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ (where \mathbb{R}^+ denotes the set of all positive real numbers) such that $V = g(x) = \frac{1}{6}\pi d^3$ to capture the exact real-world situation. Here, g is the restriction of f to \mathbb{R}^+ .

3. To draw the graph of a function, it is only customary – but not at all necessary – to consider the x -axis to be horizontal and the y -axis to be vertical; any two lines can be used to represent the x - and y -axes. In Figure 2.13, we graphically represent the function $f : \mathbb{R} \rightarrow \mathbb{R}$ such that $f(x) = y = x$ in two different ways.

2.5.3 The Range of a Function, and Various Categories of Function

The definition of a function f from X to Y only demands that every element of X be related to some element of Y ; it does not put any restriction on the use of the elements of Y . The following definition arises from this consideration.

Definition 2.25: The *range* of a function $f : X \rightarrow Y$ is the set $\{\dots, f(x), \dots\}$ of images under the mapping, and is denoted by R_f . Naturally, $R_f \subseteq Y$; that is, the range is a subset of the codomain Y of the function. \square

In general, the range is a proper subset of the codomain; that is, $R_f \subset Y$ (Figure 2.10). However, in many important cases, $R_f = Y$. Thus the following terminology is introduced.

Definition 2.26: A function $f : X \rightarrow Y$ is called *onto* (or f is a *surjective* function or a *surjection*) if the range $R_f = Y$. In other words, for an onto function f , for every $y \in Y$, there exists at least one $x \in X$ for which $f(x) = y$. If a function is not onto, it is called *into*. \square

We can look at the function in the other way too. In general, the definition of a function allows for the possibility that several x are mapped onto the same y (Figure 2.10). We may, then, define the following notion.

Definition 2.27: A function $f : X \rightarrow Y$ is called *one-to-one* (*injective*, or 1–1) if every y in R_f is the image of exactly one $x \in X$. This means that $f(x_1) \neq f(x_2)$ if $x_1 \neq x_2$ or, equivalently, $f(x_1) = f(x_2)$ implies that $x_1 = x_2$. \square

Finally, we can consider functions that are both *one-to-one* and *onto*.

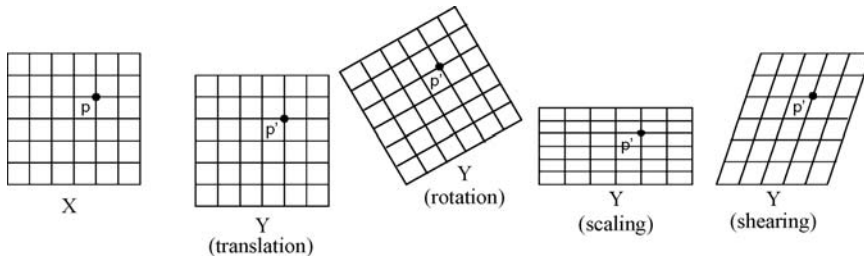


Figure 2.14 A few one-to-one onto geometric transformations in \mathbf{R}^2

Definition 2.28: A function $f : X \rightarrow Y$ is called *one-to-one onto* (or *bijective function*) if it is both one-to-one and onto. In other words, if f is a bijection, then the equation $f(x) = y$ has exactly one solution x for every $y \in Y$. Such a function is also called a *one-to-one correspondence* between X and Y . \square

Clearly, when X and Y are finite, for $f : X \rightarrow Y$ to be bijective requires that the number of elements in X be the same as the number of elements in Y .

We will now give a few examples of various types of function.

Example 2.34. We have defined the identity mapping ι of a set X as a function $f : X \rightarrow Y$ such that $x = f(x)$. This is obviously a one-to-one onto function. \square

Example 2.35. Most of the well-known geometric transformations, such as translation, rotation, reflection, scaling, shearing, and so on, are one-to-one onto functions of the form $f : \mathbf{R}^2 \rightarrow \mathbf{R}^2$ (Figure 2.14) or $f : \mathbf{R}^3 \rightarrow \mathbf{R}^3$ (see Example 2.31). \square

Example 2.36. A function $f : \mathbb{R} \rightarrow \mathbb{C}$ from the reals to the complex numbers, defined by $f(x) = i|x|$ (where $|x|$ means the absolute value of x), is neither one-to-one nor onto. However, a function $f(x) = ix$ is one-to-one but not onto. \square

Example 2.37. The perspective projection of a three-dimensional object on a two-dimensional plane is onto but not one-to-one (see Example 2.32). \square

2.5.4 Composition of Functions

In various situations, it is possible to combine two functions to define a third one.

Definition 2.29: Let $f : X \rightarrow Y$ and $g : Y \rightarrow Z$. We can define a new function, the *composite function* $g \circ f$, as follows:

$$g \circ f : X \rightarrow Z \quad \text{given by } (g \circ f)(x) = g(f(x)) \text{ for all } x \in X. \quad (2.33)$$

(Sometimes we write gf in place of $g \circ f$, and this is called the *product* of two functions f and g .) \square

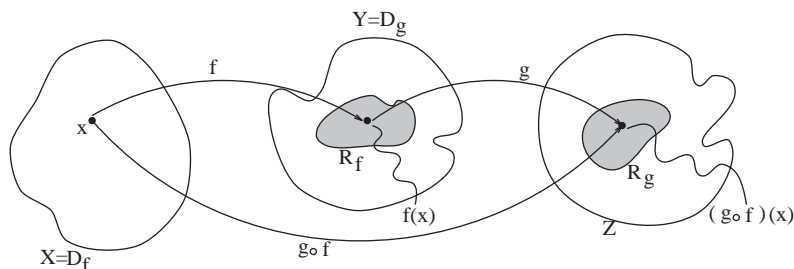


Figure 2.15 A visualization of a composite function

The notion of the composite function can be visualized as shown in Figure 2.15.

Note that in the above definition it is assumed that the range R_f of f is a subset of the domain of g , which is Y ; that is, $R_f \subseteq D_g$ – otherwise, $g \circ f$ is empty.

You should also be aware that, in general, $g \circ f \neq f \circ g$. In fact, the existence of $g \circ f$ may not even guarantee that $f \circ g$ exists. For $f \circ g$ to exist, it is necessary that $R_g \subseteq D_f$.

We will clarify the notion of the composition of two functions by means of a few examples.

Example 2.38. Given $f : \mathbb{R} \rightarrow \mathbb{R}$ such that $f(x) = x^2$ and $g : \mathbb{R} \rightarrow \mathbb{R}$ such that $g(x) = \sin x$. The composition functions are $(g \circ f)(x) = \sin(x^2)$ and $(f \circ g)(x) = (\sin x)^2$. In this case, $g \circ f \neq f \circ g$. \square

Example 2.39. Let \mathbb{Z} be the set of integers, let Y be the set $\mathbb{Z} \times \mathbb{Z}$, and suppose that $f : \mathbb{Z} \rightarrow Y$ is defined by $f(z) = (z - 1, 1)$, for all $z \in \mathbb{Z}$. Suppose that another function $g : Y \rightarrow \mathbb{Z}$ is defined by $g(y) = g(z_1, z_2) = z_1 + z_2$. In this case, $g \circ f : \mathbb{Z} \rightarrow \mathbb{Z}$, whereas $f \circ g : Y \rightarrow Y$; even to speak about the equality of $g \circ f$ and $f \circ g$ makes no sense, since they do not act on the same type of set. It is easy to see that $(g \circ f) = z$ – that is, the identity mapping of \mathbb{Z} – whereas $(f \circ g)(z_1, z_2) = (z_1 + z_2 - 1, 1)$, certainly not an identity mapping of Y ; it is not even an onto mapping of Y . \square

Example 2.40. Consider two common geometric transformations; namely, translation (denoted by α) and rotation (denoted by ρ) in \mathbb{R}^2 . It is easy to see that, in general, $\alpha \circ \rho \neq \rho \circ \alpha$. In other words, if we translate an object by some distance d and then rotate it by a certain angle θ , we do not get the same result as is obtained by first rotating the object by θ and then translating it by d . In Figure 2.16, we find that the positions of the two resulting objects are different with respect to some fixed coordinate system (the point o being taken as the pivot point of rotation). \square

The following can be easily proved for composition of functions.

Theorem 2.2. If $f : X \rightarrow Y$, $g : Y \rightarrow Z$, and, $h : Z \rightarrow W$, then

$$h \circ (g \circ f) = (h \circ g) \circ f \quad (2.34)$$

Thus the composition of the function is *associative*, and we can drop the parentheses in writing the composite function; that is, $h \circ g \circ f = h \circ (g \circ f) = (h \circ g) \circ f$.

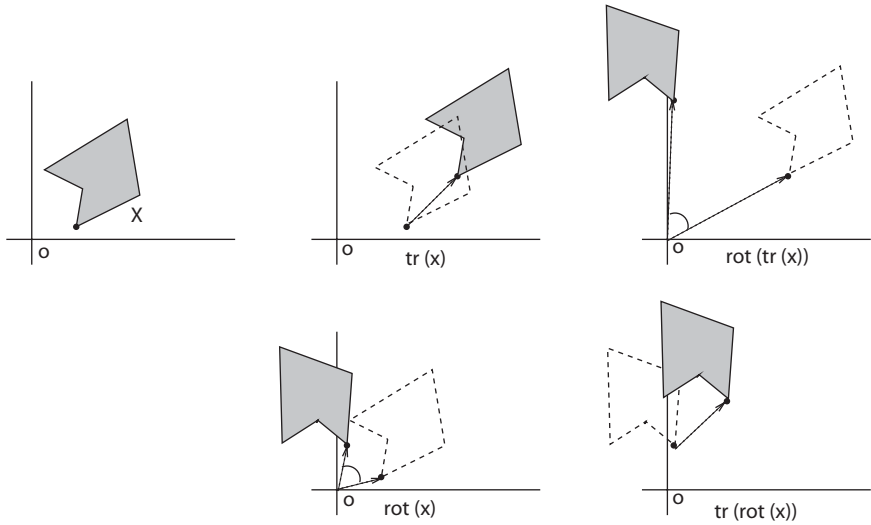


Figure 2.16 In general, *translation* \circ *rotation* is not equal to *rotation* \circ *translation*

2.5.5 The Inverse Function

The concept of the *inverse function* is very important, particularly in the field of shape description and analysis. We will introduce the idea by means of a few examples.

Example 2.41. Let X be the set of n horizontal line segments, as shown on the left side of Figure 2.17. We construct n circular regions by rotating each of the line segments on the plane; the pivot point is the midpoint of the segment. Let this set of circular regions be denoted by

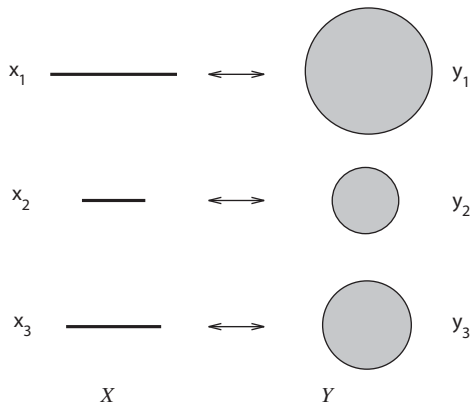


Figure 2.17 The mapping of line segments onto circular regions, and vice versa

Y (see the right side of Figure 2.17). This construction process can be defined as a function $f : X \rightarrow Y$, the rule being “assign to each line segment a circular region whose diameter is equal to the line segment.” It is easy to see that the function f is a one-to-one onto function.

Now assume that we have been given the set Y . We have to find out, for any $y \in Y$, the corresponding x ; that is, the x such that $y = f(x)$. To achieve this goal, we can define another function $g : Y \rightarrow X$ by the rule “assign to each circular region a horizontal line segment that is equal to the diameter of the circular region.” It is easy to see that for any $y \in Y$, $g(y) = x$, $x \in X$, such that $f(x) = y$. We can call the function g the *inverse function* of f and denote it by the symbol f^{-1} . Note that the inverse function f^{-1} (i.e., g) is also one-to-one onto. \square

Example 2.42. Let X be the set of all polygons in the plane and let Y be the set of all convex polygons. Let a function $f : X \rightarrow Y$ be defined by “ Y is the convex hull of X .” (The *convex hull* of a set of points A is the boundary of the smallest convex region containing A . In Figure 2.18(a), we show a polygon and its corresponding convex hull. Note that the convex hull of a convex polygon is the convex polygon itself.) It is not difficult to see that the function f is an onto function, but not one-to-one.

Is it possible to define the inverse function in this case? In other words, given any convex polygon y from the set Y , can we find the polygon x whose convex hull is y ? Clearly, such a function cannot exist, since for any convex polygon y there exist infinitely many polygons whose convex hulls will be y (Figure 2.18(b)). \square

Example 2.43. Let \mathbb{R} be the set of real numbers and let a function $f : \mathbb{R} \rightarrow \mathbb{R}$ be defined by $f(x) = e^x$, for every $x \in \mathbb{R}$. This function is one-to-one, but not onto, since the range of f excludes 0 and the negative real numbers (which means that e^x is not equal to 0 or negative

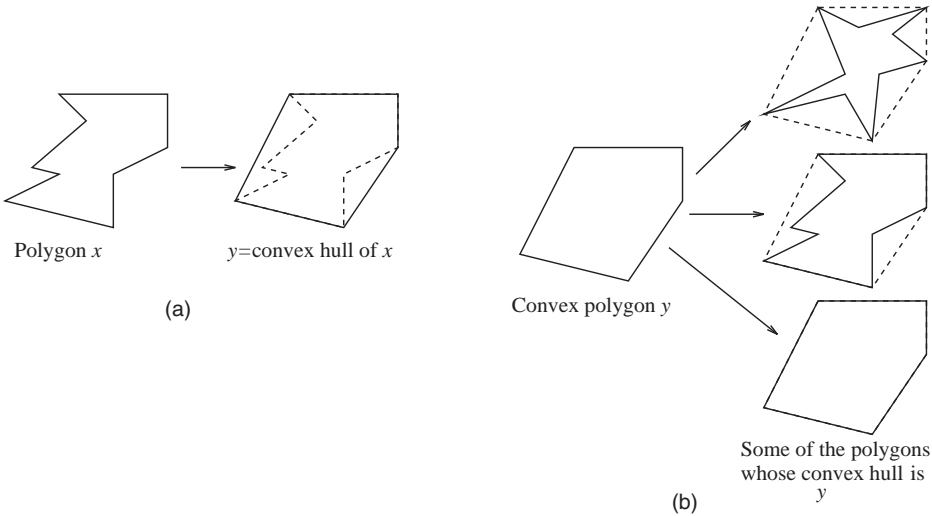


Figure 2.18 The mapping of a polygon onto its convex hull; in this case, the inverse function does not exist

for any x). Clearly, we cannot define the inverse function in this case, since for the value 0 and the negative real numbers, no corresponding image exists.

(However, if we define the same f from \mathbb{R} to \mathbb{R}^+ , where \mathbb{R}^+ denotes positive real numbers, then the inverse function f^{-1} can be defined as follows: $f^{-1} : \mathbb{R}^+ \rightarrow \mathbb{R}$, where $f^{-1}(x) = \log_e x$, for all $x \in \mathbb{R}^+$.) \square

From these examples, we can come to the following conclusion:

If and only if $f : X \rightarrow Y$ is a one-to-one and onto function, we can define an inverse function f^{-1} from Y to X . Conceptually, f^{-1} is like undoing the effect of f . If f^{-1} exists, then f is called *invertible*. Obviously, f^{-1} is also one-to-one and onto.

Below, we will present some important results concerning inverse functions.

Theorem 2.3. *The inverse of a composite function can be expressed in terms of the composition of the inverses in the reverse order; that is,*

$$(g \circ f)^{-1} = f^{-1} \circ g^{-1}. \quad (2.35)$$

We assume that both f and g are one-to-one and onto.

Theorem 2.4. (a) *If $f : X \rightarrow Y$ is a one-to-one and onto function, then $(f^{-1} \circ f) : X \rightarrow X$ is the identity function I_X and $(f \circ f^{-1}) : Y \rightarrow Y$ is the identity function I_Y .*

The converse of this theorem is as follows:

(b) *Let $f : X \rightarrow Y$ and $g : Y \rightarrow X$. Then $g = f^{-1}$ if $(g \circ f) : X \rightarrow X$ is I_X and $(f \circ g) : Y \rightarrow Y$ is I_Y .*

Let us also mention that the idea of the inverse function gives rise to the notion of *permutation*.

Definition 2.30: If the set X is finite, an invertible function from X to X – that is, $f : X \rightarrow X$ – is called a *permutation*.

In other words, any one-to-one mapping of a finite set X onto itself is called a *permutation* of X . \square

Example 2.44. If $X = \{1, 2, 3, 4, 5\}$, then one permutation might be the function f_1 such that

$$f_1(1) = 2; \quad f_1(2) = 3; \quad f_1(3) = 4; \quad f_1(4) = 5; \quad f_1(5) = 1.$$

Another permutation might be the function f_2 with

$$f_2(1) = 2; \quad f_2(2) = 3; \quad f_2(3) = 1; \quad f_2(4) = 5; \quad f_2(5) = 4. \quad \square$$

Permutations constitute a very important set of functions. For example, the role of permutation is indisputable in the study of *symmetry* of geometric objects, since the symmetry operation can be viewed as a permutation. We will discuss these points in more detail in the following chapters.

2.5.6 The One-to-One Onto Function and Set Isomorphism

We have seen that a one-to-one onto function gives rise to the notion of the inverse function. A one-to-one onto function brings out another important concept; namely, the concept of an *isomorphism*.

Definition 2.31: Suppose that there are two given sets, X and Y , and there exists a one-to-one and onto function $f : X \rightarrow Y$. We then say that the two sets X and Y are *isomorphic*, or that f furnishes an isomorphism between X and Y . \square

The importance of this notion hinges on the fact that, when two sets are isomorphic, they “behave” identically. By identical behavior we mean that the *set-theoretic behavior* of two isomorphic sets will be the same. For example: if X and Y are isomorphic sets, both of them have the same number of elements (this is routinely used to determine the size of some given set by mapping it to some set of known size); if A is a subset of X , then its corresponding image A_f must be a subset of Y ; if two subsets of X intersect, then the corresponding images in Y also intersect; and so on.

In fact, representation or modeling of objects depends heavily on the concept of set isomorphism, because the only difference between isomorphic sets lies in the concrete nature of their elements.

Note: Set isomorphism is an equivalence relation in the class of all sets.

Below, we provide a few examples of isomorphism of sets, which has numerous applications in various fields.

Example 2.45. Consider the following two sets:

$X =$ the set of points on a line L ,

$Y =$ the set of real numbers \mathbb{R} .

We all know that it is possible to define a one-to-one onto mapping between these two sets, making possible the designation of points by numbers (Figure 2.19).

At this point, we can clearly see why the Cartesian product \mathbb{R}^2 of the set of all real numbers with itself is frequently considered to be synonymous with the geometric plane \mathbf{R}^2 (see Section 2.3.8). It is simply because these two sets are isomorphic. Because of this isomorphism, at every point $p \in \mathbf{R}^2$ we can assign an ordered pair (x, y) of real numbers. In the same spirit, we know that \mathbb{R}^3 and the set of all points of our perception (the three-dimensional geometric space) \mathbf{R}^3 are isomorphic. \square

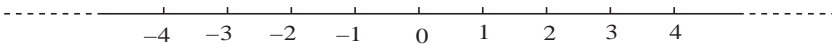


Figure 2.19 Establishing isomorphism between a set of points on a line L and the set of real numbers \mathbb{R} ; by selecting, on the line L , a fixed point as the zero-point o (the origin) and then choosing a unit of measurement, we assign to a real number r of \mathbb{R} the point of L whose distance from the origin is r units

Example 2.46. Plücker's theory of *geometric dimensionality* is very relevant at this point. We say that a geometric plane is *two-dimensional* because precisely two real numbers are necessary and sufficient to locate any particular *point* in the plane. In the same way, the space of our perception is *three-dimensional*. The kernel of Plücker's theory is that "the dimensionality of given space is not an absolute constant, but depends upon the *elements*, accepted as irreducible, in terms of which the space is described."

Let us take a simple example. Precisely *two* real numbers are necessary and sufficient to identify any particular *straight line* in a geometric plane \mathbf{R}^2 . If the equation of a straight line is $y = ax + b$, we can say that the *coordinates* of the line are (a, b) . We can, therefore, say that *a geometric plane is two-dimensional in both lines and points when either are taken as the irreducible elements out of which the plane is composed and from which the plane geometry is to be constructed.*¹ (Readers who have used the *Hough transform* in image segmentation will already be familiar with this concept.)

If, instead of points or lines, we choose circles as the irreducible elements, the geometric plane \mathbf{R}^2 becomes *three-dimensional*, because it takes precisely three numbers to specify a particular circle in the plane – two for the coordinates of the center of the circle and one for its radius. Similarly, a geometric plane is *five-dimensional* in terms of conics.

In the same vein, our space of perception \mathbf{R}^3 is *four-dimensional* in terms of straight lines as well as in terms of spheres. \square

Example 2.47. The isomorphism between two polyhedra has been established by Coxeter [14] in the following way. A polyhedron can be described "abstractly" by assigning symbols to its vertices and writing down the cycles of vertices that belong to the various faces. For instance, the abstract description of the cube as shown in Figure 2.20(a) is "1432; 2358; 3465; 5678; 6417; 7128." Two polyhedra are said to be *isomorphic* if we can find a one-to-one onto function between their abstract descriptions. For example, a cube and a parallelepiped are isomorphic (Figure 2.20(a)).

This concept of isomorphism can be extended to more complex cases. Note, for example, that a pentagon and a five-pointed, star-shaped polygon are isomorphic (Figure 2.20(b)). In the same spirit, a dodecahedron and a great stellated dodecahedron are also isomorphic (Figure 2.20(c)).

Two isomorphic polyhedra share many common properties. Clearly, they are *topologically equivalent*. As a consequence, they have the same *genus* (i.e., the same number of holes), their *reciprocals* (i.e., if the original polyhedron has n_0 vertices, n_1 edges, and n_2 faces, its reciprocal has n_2 vertices, n_1 edges, and n_0 faces) are likewise isomorphic, and so on (for more details, see Coxeter [14]). \square

2.5.7 Equivalence Relations and Functions

So far, we have seen the usefulness of the one-to-one onto function. But even if a function is not a one-to-one onto, it can give rise to interesting concepts. Here, we present an example to illustrate the point.

¹ The two-dimensionality of a geometric plane in points and in lines indicates the *principle of duality* in plane projective geometry, whereby from a statement concerning points (or lines) a *dual* statement concerning lines (or points) can be immediately inferred without independent proof.

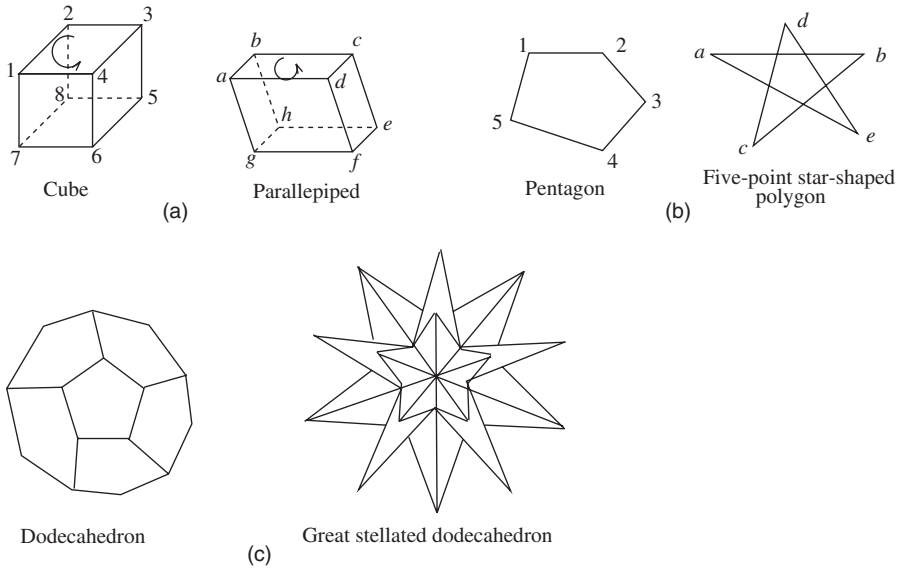


Figure 2.20 A few examples of isomorphic polygons and polyhedra

Let $f : X \rightarrow Y$. We can define in X a relation R by the following rule: aRb if and only if $f(a) = f(b)$. It is a trivial matter to check that R is an equivalence relation. Thus we have the following result.

Theorem 2.5. *Any given function f defines an equivalence relation in its domain. Equivalent elements are those whose images under f coincide.*

The equivalence relation generated by a given function f is called the *equivalence kernel* of f and is denoted by $\ker(f)$.

The converse of Theorem 2.5 is also true. Suppose that we are given an equivalence relation E in some set X . We recall (see Section 2.4.3) that it gives rise to the partition

$$X/E = \{\dots, [x], \dots\},$$

whose elements are the equivalence classes $[x]$. Let us assign to each $x \in X$ the equivalence class $[x]$ to which it belongs. In this manner, we define a function

$$p_E : X \rightarrow X/E, \quad \text{given by } p_E(x) = [x]. \quad (2.36)$$

This function is so important to the mathematician that it has a special name: it is called the *canonical mapping* or *canonical projection* of X onto X/E . Obviously, p_E is an onto function, since every $[x]$ is the image of at least one element x . Thus we have the following.

Theorem 2.6. *Any given equivalence relation E in X defines an onto function $p_E : X \rightarrow X/E$, given by $p_E(x) = [x]$.*

The ideas described above are simple but, as we shall see in much of our later work, have profound consequences. We will now present a few elementary examples that may be illustrative.

Example 2.48. Let $X = T$ be the set of all triangles in the plane and let $Y = C$ be the set of all circles in the plane. Define a function $f : T \rightarrow C$ by the rule “assign to each triangle the corresponding inscribed circle” (Figure 2.9(b)). According to Theorem 2.5, all triangles having the same inscribed circle are equivalent. \square

Example 2.49. Exploring the properties of geometric objects belonging to the same equivalence class $[x]$ is of utmost importance in geometry. Consider, for example, the set K_n of all n -sided convex polygons in the plane. Let \mathbb{R}^+ be the set of all positive real numbers. We define the function $f : K_n \rightarrow \mathbb{R}^+$ by the rule “assign to each n -sided convex polygon its perimeter.” This function gives rise to an equivalence relation among K_n : all n -sided convex polygons with the same perimeter are equivalent. We can now proceed to prove properties such as the following:

Of all n -sided convex polygons with the same perimeter, the *regular* n -sided polygon has the greatest area.

A generalization of the above result is well known:

The circle has a greater area than any other two-dimensional figure with the same perimeter. \square

2.5.8 Functions of Many Variables, n -ary Operations

So far, we have discussed functions from a set X to a set Y that are defined in the form $f(x) = y$. But there is no need to assume that our functions are limited to only one variable. For example, we often come across “a function of two variables.” This is merely a shorthand notation for the following expression:

$$f : X \times Y \rightarrow Z \quad \text{being given by some rule } f(x, y) = z. \quad (2.37)$$

Functions of several variables can be defined in a similar way.

For functions of many variables, the following two special cases are of importance:

- An important class of such functions is the following. Let

$$p_{proj} : X \times Y \rightarrow X \quad \text{being given by } f(x, y) = x. \quad (2.38)$$

We call p_{proj} a *projection* onto the X -axis. Similarly, we can define a projection onto the Y -axis.

- It is mentioned in Section 2.5.1 that the term “operation” is used in special cases as a synonym for “function.” A formal definition of “operation” can be given at this point in terms of functions of many variables.

Definition 2.32: Let X be a set and f be a mapping $f : X \times X \rightarrow X$. Then f is called a *binary operation* on X . In general, a mapping $f : X \times X \times \cdots (n\text{-terms}) \rightarrow X$ – that is,

$f : X^n \rightarrow X$ – is called an *n*-ary operation and *n* is called the *order* of the operation. For $n = 1$, $f : X \rightarrow X$ is called a *unary* operation. \square

If a function on the members of a set produces images that are also members of the same set, then that set is said to be *closed* under that function and this property is called the *closure* property. The definition of *n*-ary operations implies that the sets on which such operations are defined are closed under these operations. This property distinguishes the *n*-ary operations from other functions².

2.5.9 A Special Type of Function: The Analytic Function

Definition 2.33 (analytic function): A function $f(x)$ is real analytic on an open set D in the real number domain of x if, for any $x_0 \in D$, we can write

$$f(x) = \sum_{n=0}^{\infty} a_n(x - x_0)^n \quad (2.39)$$

$$= a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + a_3(x - x_0)^3 + \dots \quad (2.40)$$

where a_0, a_1, a_2, \dots are real numbers and the series is convergent for x in a neighborhood of x_0 . \square

Alternatively, this defines that an analytic function is an infinitely differentiable function such that the Taylor series at any point x_0 in its domain,

$$T(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n, \quad (2.41)$$

is convergent for x close enough to x_0 and its value equals $f(x)$.

The definition of a complex analytic function is obtained by replacing all instances of “real” above with “complex.”

For example, any polynomial (real or complex) is an analytic function. The exponential function, the trigonometric functions, the logarithm, and the power functions are also analytic on any open set of their domain.

The properties of analytic functions are as follows. The sums, products, and compositions of analytic functions are also analytic. The reciprocal of an analytic function that is nowhere zero is analytic, as is the inverse of an invertible analytic function whose derivative is nowhere zero. Any analytic function is smooth.

Any analytic function (real or complex) is differentiable – actually, infinitely differentiable (that is, smooth). But there exist smooth real functions that are not analytic. The real analytic functions are much “fewer” in number than the real (infinitely) differentiable functions.

However, the situation is quite different for complex analytic functions. It can be proved that any complex function differentiable in an open set is analytic. Consequently, in complex analysis, the term “analytic function” is synonymous with *holomorphic* function.

² In this book, we use the term “operation” as a synonym for “function,” and do not restrict ourselves to Definition 2.32.

Real and complex analytic functions have important differences, which result mainly from their different relationship with differentiability. Complex analytic functions are in many ways more rigid.

Analytic functions in several variables can be defined by means of power series in those variables, and have some of the same properties as analytic functions of one variable. However, especially for complex analytic functions, new and interesting phenomena show up when working in two or more dimensions. These result from the fact that, for the analytic functions of several variables, the so-called *factorization theorem*, which tells us that if a single variable function $f(x) = 0$ at $x = x_0$, then it has a factor of $(x - x_0)$, does not hold.

3

Algebraic Structures for Shape Description

3.1 What is an Algebraic Structure?

Assume that we are given a set A that consists of five rectangles, as shown in Figure 3.1.

As such, the set conveys very little meaning to us; it is just a collection of five arbitrary rectangles.

But if we are made aware that those five rectangles are not completely arbitrary, that there is a certain relationship among the elements, the set A becomes more meaningful; the set A can no longer be viewed as a mere collection of rectangles, but it becomes a structured set with more specific properties. The condition that holds among the elements of A , in this particular case, is the following: “take any pair of elements, translate them to coincide their midpoints (shown as a dot in every case), and take their intersection. The resulting polygon will be one of the elements of A .” It is not difficult to see that this rule on the elements of A is simply a function

$$f : A \times A \rightarrow A \quad \text{given by } f(a, b) = c,$$

where $a, b, c \in A$.

Such a function f is a special function, since any element c of A is a composite of two elements, a and b , of A . Therefore, such a function is called an *algebraic composition law*. Instead of writing $f(a, b) = c$, it might be more convenient in such a situation to devise some special symbol – say, \square – to denote this composition. Thus we can express this function as $a \square b = c$, just as we express $2 + 3 = 5$ in ordinary algebra. In this particular example, the composition law can be completely specified by the following table:

\square	a_1	a_2	a_3	a_4	a_5
a_1	a_1	a_3	a_3	a_1	a_3
a_2	a_3	a_2	a_3	a_5	a_5
a_3	a_3	a_3	a_3	a_3	a_3
a_4	a_1	a_5	a_3	a_4	a_5
a_5	a_3	a_5	a_3	a_5	a_5

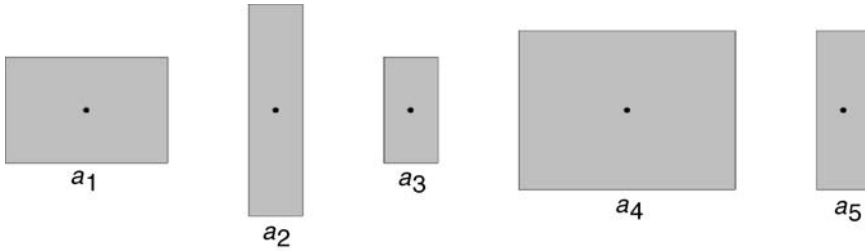


Figure 3.1 A set $A = \{a_1, a_2, a_3, a_4, a_5\}$ whose elements are rectangles

Some of our observations from the above example can be generalized in the following way:

- (a) A structure of a specific kind can be imposed on a set A by defining an *algebraic composition law* on A .
- (b) An algebraic composition law on A specifies a way in which two elements are “combined” to yield a third element of the set A . Thus a composition law is a special kind of function $f : A \times A \rightarrow A$. In fact, a composition law can be generalized as a function

$$g : A \times A \times \cdots \times A \rightarrow A; \quad \text{that is, } A^n \rightarrow A,$$

which could be considered as an n -ary composition law. Thus $f : A \times A \rightarrow A$ is a *binary* composition law, $h : A \rightarrow A$ is a *unary* composition law, and so on. However, binary composition laws are so frequently in use that by “composition law,” without any qualifying adjective, we generally mean binary composition law.

- (c) Instead of writing in the functional notational form – that is, in the form $f(a, b) = c$ – we can devise some special symbol, such as Δ , \square , $+$, $*$, \times , \uplus , and so on to denote an algebraic composition law, say as $a * b = c$. (In some of the literature, an algebraic composition law is called an “operation,” to distinguish it from the general class of functions. However, since “function” and “operation” are considered as synonyms in this book, we use “composition law” to emphasize its distinctiveness.)
- (d) Since, by means of an algebraic composition law, the members of a set produce images that are also members of the same set, the set is said to be *closed* under that composition law (see Section 2.5.8).
- (e) Further generalization of the idea of composition law is possible. Let A (with elements a, b, \dots) and B (with elements α, β, \dots) be two *different* sets. Then a function

$$k : B \times A \rightarrow A \quad \text{given by } k(\alpha, a) = b,$$

is also a composition law. To distinguish it from the previous composition law, where only one set A is involved, we call this composition law an *external* composition law. In contrast to that, the former one could be called an *internal* composition law. There is no real difference between internal and external composition laws, except for practical considerations – the former is a special case of the latter.

Now we can define what an algebraic structure is.

Definition 3.1: A set together with at least one algebraic composition law is called an *algebraic structure* or *algebraic system*. (By definition, the set must be closed under the composition law(s).) \square

An algebraic structure on a set A defined by an internal composition law $*$ is denoted by the symbol $(A, *)$. A given set A may, of course, carry different algebraic structures and we distinguish them by writing $(A, *)$, $(A, +)$, (A, \cdot) , and so on, in each case using a different symbol for the appropriate composition law. It often happens that a fixed set is endowed with two (or more) distinct composition laws and that these composition laws are related to each other. In that case, we write $(A, *, +)$ and so on. If the composition law is external, then we can use the notation $(A, B, *)$.

Note: From the *computational* viewpoint, one immediate advantage of an algebraic system could be mentioned here. It is often possible to start with only a few elements of a set, and then repeatedly apply the algebraic composition law(s) of the system on those elements and their composites to produce all other elements of the set. Consider, for instance, the example shown in Figure 3.1. It is possible in that case to start with the elements a_1 , a_2 and a_4 of the set to produce the other two elements a_3 and a_5 . A subset whose compositions (with repetitions) comprise the whole set can be called a set of *generators*. This idea comes very close to the *genetic method* of describing shapes (see Section 1.4.3).

Below, we present a few examples of familiar algebraic structures.

3.1.1 Algebraic Systems with Internal Composition Laws

Example 3.1. Let \mathbb{R} be the set of all real numbers and let $+$ be ordinary arithmetic addition. If we add any two real numbers, the result will be another real number. So $+$ denotes an internal composition law and $(\mathbb{R}, +)$ is an algebraic system. Similarly, if \cdot represents ordinary arithmetic multiplication, (\mathbb{R}, \cdot) is another algebraic system. Moreover, the addition and multiplication operations are related, since

$$a \cdot (b + c) = a \cdot b + a \cdot c, \quad (b + c) \cdot a = b \cdot a + c \cdot a,$$

where $a, b, c, \in \mathbb{R}$.

We can say that $(\mathbb{R}, +, \cdot)$ is also another algebraic system.

In a similar way, if \mathbb{N} , \mathbb{N}^+ and \mathbb{Z} denote the sets of natural numbers (the positive integers) and integers, respectively, then $(\mathbb{N}, +)$, (\mathbb{N}, \cdot) , $(\mathbb{N}, +, \cdot)$, $(\mathbb{N}^+, +)$, (\mathbb{N}^+, \cdot) , $(\mathbb{N}^+, +, \cdot)$, $(\mathbb{Z}, +)$, (\mathbb{Z}, \cdot) , $(\mathbb{Z}, +, \cdot)$, are all algebraic systems. These are the trivial algebraic systems and we all are aware of them.

But consider the set of all odd integers O . While (O, \cdot) is an algebraic system, $(O, +)$ is not. On the other hand, if E is the set of all even integers, both $(E, +)$ and (E, \cdot) are algebraic systems. \square

Example 3.2. Let K be the set of all convex subsets of the three-dimensional geometric space; that is, of \mathbb{R}^3 . If the mapping of any two members a, b of K is $c = a \cap b$ (where \cap denotes set intersection), then c also belongs to K , since the intersection of two convex sets is also a convex set. Thus the mapping is from $K \times K$ to K , and is an algebraic composition law. Therefore, (K, \cap) forms an algebraic system. \square

Note: The central idea of this particular algebraic system is frequently employed in various shape description schemes in different forms. The advantage is that starting with simple convex shapes, and repeatedly applying the composition \cap , very many complex shapes can be easily produced. *Half-spaces* are often chosen as simple convex shapes. (The description of a half-space goes as follows. Define any plane in \mathbf{R}^3 , it divides the whole three-dimensional space into two halves; depending on the chosen convention, one half is called the *left half-space* and the other the *right half-space*. It is not difficult to see that a half-space is a convex set.) By means of intersection of half-spaces, we can produce any convex shape in \mathbf{R}^3 . The analogous method in \mathbf{R}^2 is presented in Section 1.4.3 to describe convex polygons in the plane.

Example 3.3. Let F denote the set of all functions from a set X into X ; that is, $F = \{f \mid f : X \rightarrow X\}$. Let us define a function h as follows:

$$h : F \times F \rightarrow F \quad \text{given by } h(f_1, f_2) = f_1 \circ f_2,$$

where $f_1, f_2 \in F$ and \circ denotes the composition of the function (see Section 2.5.4). Clearly, h defines a composition law on F . We can, therefore, say that (F, \circ) forms an algebraic system.

(F, \circ) is a very general algebraic system, and we find that various instances of the system are frequently being used. To clarify the reason for this, let us first consider a simple instance of the system.

Let $X = \{a, b\}$ and $F = \{f_1, f_2, f_3, f_4\}$, where

$$\begin{aligned} f_1(a) = a, \quad f_1(b) = b; & \quad f_2(a) = a, \quad f_2(b) = a; \\ f_3(a) = b, \quad f_3(b) = b; & \quad f_4(a) = b, \quad f_4(b) = a. \end{aligned}$$

The composition table for \circ in this case is given below:

\circ	f_1	f_2	f_3	f_4
f_1	f_1	f_2	f_3	f_4
f_2	f_2	f_2	f_2	f_2
f_3	f_3	f_3	f_3	f_3
f_4	f_4	f_3	f_2	f_1

In geometry, we come across many special instances of this system. Let us mention a few of them here:

1. Let us consider the set of all geometric transformations on the plane (i.e., $X = \mathbf{R}^2$ in this case) that are one-to-one and onto: call this set \mathcal{G} (instead of F). Every translation, rotation, scaling, shearing, reflection, and so on are some of the elements of \mathcal{G} . Obviously, there are many other transformations that belong to \mathcal{G} . In simple terms, if g is an element of \mathcal{G} , then for every point p in the plane \mathbf{R}^2 there is a unique point q in \mathbf{R}^2 such that $g(p) = q$ and, conversely, for every point r in the plane there is a unique point s in the plane such that $r = g(s)$. It is easy to see that if $g_1, g_2 \in \mathcal{G}$, then $g_1 \circ g_2$ is also an element of \mathcal{G} ; that is, $g_1 \circ g_2$ is also a one-to-one onto geometric transformation on the plane. For example, a combination of any translation and rotation is another element of \mathcal{G} . Thus (\mathcal{G}, \circ) forms an algebraic system. (Sometimes, $g_1 \circ g_2$ is called the *product* of transformations instead of the *composition* of transformations.)
2. The set \mathcal{G} of all one-to-one onto geometric transformations on the plane is too large to be very interesting. We can consider some smaller subsets of \mathcal{G} . One interesting subset is the

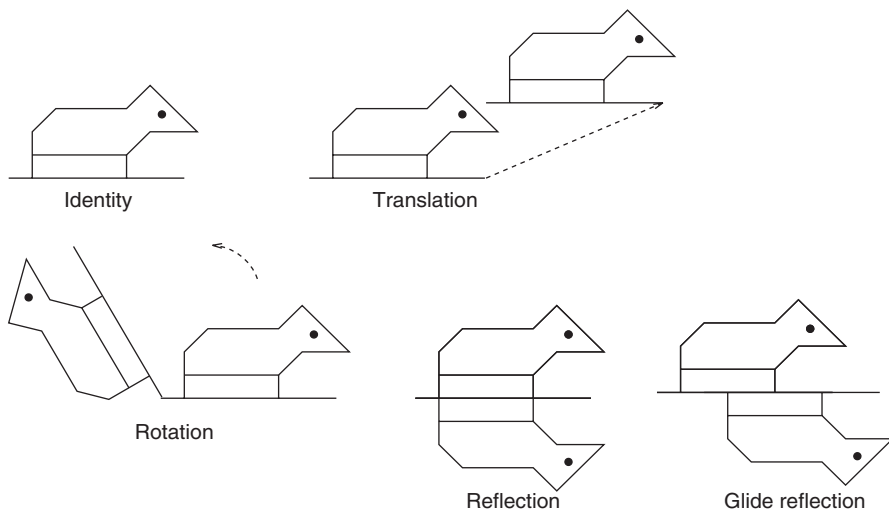


Figure 3.2 The isometries on the plane

set of all *congruent transformations* on the plane. In the mathematical literature, congruent transformations are also called *isometries*. A geometric transformation α is an isometry if $|p'q'| = |pq|$ for all points p and q in the plane, where $p' = \alpha(p)$ and $q' = \alpha(q)$; by the term $|p'q'| = |pq|$, we mean that the distance from p' to q' is equal to the distance from p to q . This means that “isometry” is a geometric transformation that “preserves distance.” The term comes from the Greek words “*isos*” (equal) and “*metron*” (measure). The set of all isometries on the plane is generally denoted by the symbol \mathcal{I} . The set \mathcal{I} contains the identity transformation, all translations, all rotations, all reflections, and all glide reflections on the plane (Figure 3.2). If there is an isometry that transforms one figure into another figure, we say that these two figures are congruent. This is why isometry is also known as congruent transformation.

It is easy to see that (\mathcal{I}, \circ) is an algebraic system, since the composition of any two isometries is another isometry. \square

In our subsequent discussions, we shall come across many more algebraic systems with internal composition laws.

3.1.2 Algebraic Systems with External Composition Laws

Example 3.4. Let \mathbb{R}^+ be the set of all positive real numbers and let \mathbb{N} be the set of all natural numbers. Consider the mapping of $\mathbb{N} \times \mathbb{R}^+$ that assigns to every couple (n, r) the positive real number r^n , where $n \in \mathbb{N}$ and $r \in \mathbb{R}^+$. This is an external composition law for \mathbb{R}^+ , commonly known as *exponentiation*. \square

Example 3.5. Let \mathbb{C} be the set of all complex numbers of the form $a + ib$, where $a, b \in \mathbb{R}$, the set of real numbers. Consider the mapping of $\mathbb{R} \times \mathbb{C}$ into \mathbb{C} and assign to every couple (r, c)

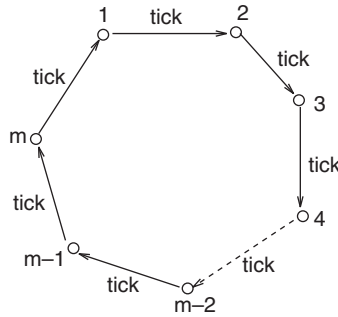


Figure 3.3 A clock algebra

the complex number $c' = ra + irb$, where $a + ib = c \in \mathbb{C}$. This is an external composition law for \mathbb{C} . If we take the geometrical interpretation of a complex number $a + ib$ as the point in \mathbb{R}^2 that can be coordinatized as (a, b) , or the vector from the origin to the point (a, b) , the above function is then called *scalar multiplication*. \square

Although binary composition laws are most common in everyday practice, we also come across n -ary composition laws, where $n \neq 2$. Consider the following example.

Example 3.6. Let $M = \{1, 2, \dots, m\}$ and let *tick* be a unary composition law on M , given by

$$\text{tick}(j) = \begin{cases} j + 1 & j \neq m, \\ 1 & j = m. \end{cases}$$

The algebraic system (M, tick) is called a *clock algebra*, for obvious reasons. The result of the composition law *tick* on the elements of M is illustrated in Figure 3.3. It is easy to see that $1 \in M$ is the *generator* of the system (M, tick) . \square

We have mentioned that a fixed set may be endowed with two or more distinct composition laws that are also related to each other. We will now present a few examples of such systems.

Example 3.7. The first examples are from the number systems:

1. The algebraic systems $(\mathbb{R}, +, \cdot)$, $(\mathbb{Z}, +, \cdot)$, and so on (mentioned in Example 3.1) are clearly systems of this kind.
2. It is not difficult to find systems of this type that are related to shapes of objects. Let \mathbf{R}^3 be the three-dimensional geometric space and let $\mathcal{P}(\mathbf{R}^3)$ be its power set; that is, all the shapes in \mathbf{R}^3 that we can imagine. Let us consider two composition laws: set union \cup and set intersection \cap . $(\mathcal{P}(\mathbf{R}^3), \cup, \cap)$ is obviously an algebraic system with two internal composition laws. The basic idea of this particular algebraic system is used in many existing shape description systems.
3. In fact, it is possible to envisage a more complex algebraic system than the one just mentioned. As before, let $\mathcal{P}(\mathbf{R}^3)$ be the power set of \mathbf{R}^3 . For any sets $A, B \in \mathcal{P}(\mathbf{R}^3)$, define

two composition laws \uplus and \odot on $\mathcal{P}(\mathbf{R}^3)$ as

$$A \uplus B = (A \setminus B) \cup (B \setminus A),$$

$$A \odot B = A \cap B,$$

where \setminus denotes the set difference operation.

It can easily be verified that, for any $A, B, C \in \mathcal{P}(\mathbf{R}^3)$,

$$A \odot (B \uplus C) = (A \odot B) \uplus (A \odot C);$$

that is, the operation \odot distributes over \uplus , exactly in the way that the ordinary multiplication operation \cdot distributes over the addition operation $+$ in the system $(\mathbb{R}, +, \cdot)$. Thus $(\mathcal{P}(\mathbf{R}^3), \uplus, \odot)$ forms an algebraic system. \square

We conclude this general discussion of algebraic systems with a consideration concerning the *restriction of a composition law* to a subset. Take, for example, the two algebraic systems (\mathbb{R}, \cdot) and (\mathbb{N}, \cdot) . Here, $\mathbb{N} \subset \mathbb{R}$. In such a case, we can say that the algebraic system (\mathbb{N}, \cdot) is a subsystem of (\mathbb{R}, \cdot) . But this does not imply that any subset of \mathbb{R} with \cdot as the composition law is a subsystem of (\mathbb{R}, \cdot) . For example, consider the set of the first n natural numbers – say, $\mathbf{n} = \{1, 2, \dots, n\}$. Obviously, $\mathbf{n} \subset \mathbb{R}$, but (\mathbf{n}, \cdot) is not an algebraic system in its own right, because it is not closed; the product of two numbers k and l , both less than or equal to n , may or may not be less than or equal to n . Thus we arrive at the following definition.

Definition 3.2: Let $(A, *)$ be an algebraic system and let $B \subset A$. If $(B, *)$ is also an algebraic system in its own right, then it is called a *subsystem* of $(A, *)$. \square

In Example 3.3, the system (\mathcal{I}, \circ) is a subsystem of (\mathcal{G}, \circ) .

3.2 Properties of Algebraic Systems

A fundamental question arises at this point: “Even if we identify that a given set of objects possesses an ‘algebraic structure,’ how much is gained in practice from this discovery?” Of course, we get to know that the set of objects is closed under some algebraic composition law, and if it becomes possible to identify its set of generators, we can construct the whole set from that subset. But can we envisage some “stronger” structure than this? Informally, the idea of a “stronger” structure can be expressed as follows. Let the elements of a set A satisfy n conditions, and let those of another set B , in addition to those n conditions, satisfy a further m conditions; we can then say that B has a stronger structure than A . Therefore, in the case of an algebraic system, a measure of structure of the system needs to be assessed in terms of the properties of its composition laws – the stronger the properties of the composition laws become, the more structure the system gains. By “property of an algebraic system,” we mean here the properties possessed by its composition laws.

Of course, it is possible to consider various kinds of properties of an algebraic system. However, we list below only those properties that are regarded by mathematicians as fundamentals, and that are also regarded as important for practical needs.

3.2.1 Associativity

An internal composition law $*$ on A is said to be associative if

$$(a * b) * c = a * (b * c), \quad \text{for all } a, b, c \in A.$$

Most of the algebraic systems that we encounter in practice are associative. Consider, for example, the systems that we mentioned earlier – $(\mathbb{R}, +)$, (\mathbb{R}, \cdot) , $(\mathcal{P}(\mathbf{R}^3), \cup)$, $(\mathcal{P}(\mathbf{R}^3), \cap)$, (K, \cap) , (\mathcal{G}, \circ) , (\mathcal{I}, \circ) , and so on – which are all associative systems.

However, surely nonassociative algebraic systems can exist. For example, consider the set of all real numbers between 0 and 1. Denote it by A . Also consider the following composition law:

$$a * b = \min\{1 - a, b\},$$

where $a, b \in A$.

It is not difficult to check that this particular composition law is not associative. Take $a = \frac{3}{4}$, $b = \frac{1}{8}$, and $c = \frac{1}{2}$; you will find that while $(a * b) * c = \frac{1}{2}$, $a * (b * c) = \frac{1}{4}$. The most familiar class of nonassociative algebras are known as *Lie algebras* (for further details, see Roman [87]).

3.2.2 Commutativity

An internal composition law is commutative if

$$a * b = b * a, \quad \text{for all } a, b \in A.$$

Many familiar composition laws are commutative. For example, $(\mathbb{R}, +)$, (\mathbb{R}, \cdot) , $(\mathcal{P}(\mathbf{R}^3), \cup)$, $(\mathcal{P}(\mathbf{R}^3), \cap)$, (K, \cap) are commutative systems.

However, the composition law \circ in the algebraic systems (\mathcal{G}, \circ) or (\mathcal{I}, \circ) is, in general, noncommutative.

This can be demonstrated by the following simple example. We know that all reflections on the plane belong to the set \mathcal{I} (see Example 3.3). We consider two reflections, σ_a and σ_b , in lines a and b , respectively. Obviously, $\sigma_a, \sigma_b \in \mathcal{I}$. We consider the reflections of a triangle T , which is shown in Figure 3.4.

Since the lines a and b happen to be parallel, both $\sigma_a \circ \sigma_b = \sigma_a(\sigma_b(T))$ and $\sigma_b \circ \sigma_a = \sigma_b(\sigma_a(T))$ are equivalent translations. However, $\sigma_a \circ \sigma_b$ and $\sigma_b \circ \sigma_a$ do not produce the same translation, as has been clearly depicted in Figure 3.4.

It is also easy to verify that if τ represents a translation on the plane and ρ denotes rotation, then $\tau, \rho \in \mathcal{I}$. But $\tau \circ \rho \neq \rho \circ \tau$.

3.2.3 Distributivity

Consider an algebraic system $(A, *, \circ)$ with two composition laws. We say that \circ is *left distributive* over $*$ if

$$a \circ (b * c) = (a \circ b) * (a \circ c),$$

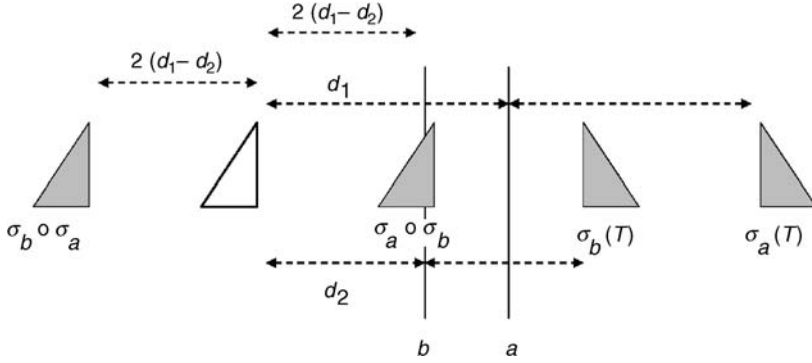


Figure 3.4 A demonstration of the fact that (\mathcal{I}, \circ) is a noncommutative system

and that \circ is *right distributive* over $*$ if

$$(a * b) \circ c = (a \circ c) * (b \circ c),$$

for all $a, b, c \in A$.

In the familiar $(\mathbb{R}, +, \cdot)$ system, multiplication is both left and right distributive over addition, since $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(a + b) \cdot c = a \cdot c + b \cdot c$, for all $a, b, c \in \mathbb{R}$. But note that addition is not distributive over multiplication, since $a + (b \cdot c) \neq (a + b) \cdot (a + c)$. On the other hand, consider the system $(\mathcal{P}(\mathbb{R}^3), \cup, \cap)$. Since $a \cup (b \cap c) = (a \cup b) \cap (a \cup c)$ and $a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$ for all $a, b, c \in \mathcal{P}$, both union and intersection of sets distribute over each other. Similarly, in the system $(\mathcal{P}(\mathbb{R}^3), \uplus, \odot)$, the operation \odot distributes over \uplus (see Example 3.7).

3.2.4 The Existence of the Identity/Unit Element

If for an internal composition law there exists a distinguished element $e \in A$ such that

$$e * a = a \quad \text{and} \quad a * e = a, \quad \text{for all } a \in A,$$

then the system is said to be equipped with the *unit element* or *identity element* e .

Note: If only one of the stated equations holds, we speak of a *left* (or *right*) identity element, respectively. This relaxed concept is not very important, so when we say “identity element,” we always assume it to be a “two-sided” identity element.

Theorem 3.1. *If a unit element exists, then it is unique.*

Proof: The validity of the result is easy to check. Suppose that e and e_1 are both unit elements. Then $e * e_1 = e_1$ and also $e * e_1 = e$, so that $e_1 = e$. \square

In the system $(\mathbb{R}, +)$, the element $0 \in \mathbb{R}$ is the identity element with respect to addition. Similarly, $1 \in \mathbb{R}$ is the identity element with respect to multiplication in the system (\mathbb{R}, \cdot) . In

the case of the system $(\mathcal{P}(\mathbf{R}^3), \cup)$, the empty set \emptyset is the identity, while the universal set \mathbf{R}^3 is the identity element in the system $(\mathcal{P}(\mathbf{R}^3), \cap)$. The same elements \emptyset and \mathbf{R}^3 are the identity elements with respect to \uplus and \odot , respectively, in the system $(\mathcal{P}(\mathbf{R}^3), \uplus, \odot)$. It is not difficult to see that the identity transformation, say ι , is the identity element with respect to product of transformation \circ in the systems (\mathcal{G}, \circ) or (\mathcal{I}, \circ) .

However, there are many algebraic systems that are not equipped with any identity element. One such system is $(\mathbb{N}^+, +)$.

3.2.5 The Existence of an Inverse Element

Suppose that an algebraic system $(A, *)$ has an identity element e . An element $a \in A$ is said to have an *inverse* if there exists an element $a' \in A$ such that

$$a' * a = e \quad \text{and} \quad a * a' = e.$$

Note: If only one of the two stated conditions holds, we call a' a *left* (or *right*) inverse of a , respectively.

If an element a has exactly one inverse, it is standard practice to denote this inverse by the symbol a^{-1} .

To give trivial examples of inverses, note that in the system $(\mathbb{R}, +)$ every element has an inverse – namely, $a^{-1} = -a$; whereas in (\mathbb{R}, \cdot) every element except the number zero has an inverse – that is, $a^{-1} = \frac{1}{a}$. In the system (\mathcal{G}, \circ) , every element $g \in \mathcal{G}$ has an inverse g^{-1} , since g is a one-to-one onto function (see Section 2.5.5). The same is true for the system (\mathcal{I}, \circ) too.

A few interesting facts regarding inverse elements can be stated in the following theorem.

Theorem 3.2. *Let $(A, *)$ be an algebraic system containing the identity element e and let the composition law $*$ be associative.*

- (a) *If an element a has an inverse a' , this is unique (i.e., a has exactly one inverse).*
- (b) *If a' is the inverse of an element a , then a is the inverse of a' .*
- (c) *If, further, b' is the inverse of an element b , then $b' * a'$ is the inverse of $a * b$.*
- (d) *The identity element e is its own inverse. (The condition of associativity of $*$ is not required in this case.)*

The proof is easy and is left to the readers.

Note: We can make one important observation explicitly here. We have already pointed out that the geometric system $(\mathcal{P}(\mathbf{R}^3), \cup, \cap)$, or some variation of it, is frequently used for the purpose of shape description. We have also shown that this particular system and the familiar real number system $(\mathbb{R}, +, \cdot)$ possess many identical properties, such as those concerning associativity, commutativity, distributivity, and the identity element. However, with regard to the question of inverse elements, they differ. The former system does not contain an inverse element for every element of $\mathcal{P}(\mathbf{R}^3)$ – neither for union \cup nor for intersection \cap . (For \cup , the only element that has an inverse is \emptyset , and its inverse is itself. For \cap , the only element that has an inverse is \mathbf{R}^3 , and its inverse is again itself.) On the other hand, in the system $(\mathbb{R}, +)$, every element has an inverse. This difference, we feel, is a crucial one.

For the time being, these five properties are sufficient to invent and/or explore structures of various algebraic systems. However, the need will arise to define more such properties and we shall do so in the appropriate places.

3.3 Morphisms of Algebraic Systems

The purpose of this section is to establish relations between different algebraic systems. It is frequently found that many seemingly different algebraic systems exhibit similar characteristics and, in fact, one system that is a little more familiar one can be used to study the others. Let us take a simple example. We all are aware of the addition and multiplication of even and odd integers, which can be expressed as follows:

$+$ (addition)	even	odd	\cdot (multiplication)	even	odd
even	even	odd	even	even	even
odd	odd	even	odd	even	odd

If we denote the set $\{even, odd\}$ by the symbol A , then $(A, +, \cdot)$ is an algebraic system.

Now consider the algebra of positive integers “modulo 2.” By “modulo m ,” we mean dividing a positive integer by m and considering only the remainder. In the case of modulo 2, the remainder is either 0 or 1. Thus its algebra can be described by the following tables:

\oplus (addition)	0	1	\odot (multiplication)	0	1
0	0	1	0	0	0
1	1	0	1	0	1

If we write $B = \{0, 1\}$, then obviously (B, \oplus, \odot) denotes this algebraic system. It can be immediately observed that the systems $(A, +, \cdot)$ and (B, \oplus, \odot) are basically the same. If we assume a function $f : A \rightarrow B$ such that $f(even) = 0$ and $f(odd) = 1$, then the composition laws $+$ and \cdot in the first system turn out to be exactly the same as \oplus and \odot in the second system. These two systems, it seems, differ only in the notation of their elements; otherwise, they are the same. In mathematical language, we call these two systems *isomorphic* (literally, of the same form).

We have seen that it is profitable to identify isomorphism between two completely unstructured sets (see Section 2.5.6). Identifying isomorphism between two algebraic systems is much more rewarding, since the two sets are more structured sets in this case and we are establishing correspondence not only between the two sets but also between two structures. No doubt one of the most important concepts of algebraic structures is that of *morphism* and, in particular, *isomorphism*.

Let us state the concept of morphism of two algebraic systems more formally.

Definition 3.3: Let $(A, *)$ and $(A', *')$ be two algebraic systems. A function $f : A \rightarrow A'$ is called a *homomorphism*, or simply a *morphism*, from $(A, *)$ to $(A', *')$ if, for any $a, b \in A$,

$$f(a * b) = f(a) *' f(b).$$

If such a function f exists, then it is customary to call $(A', *')$ a homomorphic image of $(A, *)$, although it must be noted that the range R_f of f may not be equal to A' ; that is, $R_f \subseteq A'$. \square

So, a homomorphism from one algebraic system to another is a mapping between sets with extra conditions that preserve the algebraic structure.

Homomorphisms are *classified* according to the mapping properties of the function f that provides the homomorphisms. A homomorphism is called a

monomorphism if f is *one-to-one*,

epimorphism if f is *onto*,

isomorphism if f is *one-to-one onto*.

Before proceeding further, we will present a few examples of morphisms.

Example 3.8. The function $\log : \mathbb{R}^+ \rightarrow \mathbb{R}$ defines an isomorphism of (\mathbb{R}^+, \cdot) onto $(\mathbb{R}, +)$ because $\log(a \cdot b) = \log(a) + \log(b)$. The practical use of logarithms is based on this isomorphism. \square

Example 3.9. Let us consider a set A consisting of the identity transformation ι and only one reflection, say σ , on the plane; that is, $A = \{\iota, \sigma\}$ (see Example 3.3 and Figure 3.5).

If \circ denotes the product/composition of transformations, then the composition law can be expressed as the following table:

\circ	ι	σ
ι	ι	σ
σ	σ	ι

Note that the product of two reflections is the identity; that is, $\sigma \circ \sigma = \iota$. Clearly, (A, \circ) is an algebraic system.

The system (A, \circ) is isomorphic to the system (B, \cdot) , where $B = \{1, -1\}$ and \cdot denotes ordinary multiplication.

It is not difficult to realize that the system (A, \circ) is an algebraic system and that it is isomorphic to (B, \cdot) ; the basic requirement is that $\sigma \circ \sigma = \iota$. Therefore, we can replace the reflection σ by the rotation ρ through 180° about some axis where the same condition holds; that is, $\rho \circ \rho = \iota$. (It can be mentioned here that a one-to-one onto geometric transformation γ is called an *involution* if and only if $\gamma \circ \gamma = \iota$, but $\gamma \neq \iota$. An involution can also be

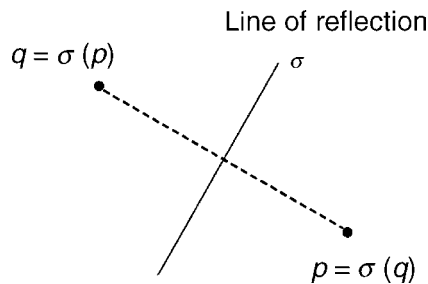


Figure 3.5 A reflection σ and the product $\sigma \circ \sigma = \iota$ of a point p in \mathbb{R}^2

characterized in the following way: a nonidentity transformation γ is an involution if and only if $\gamma = \gamma^{-1}$.) \square

Example 3.10. The idea of *symmetry* is familiar to most of us. However, not many people realize that there is a consequential *algebra of symmetry*. Let us present here a very simple algebra of symmetry, in the concrete case of the symmetries of an equilateral triangle on the plane.

When we say that a figure is “symmetrical,” we mean that there is a congruent transformation (i.e., isometry) that leaves the figure unchanged as a whole, merely permuting its component elements. Mathematically, an isometry (congruent transformation) α is a *symmetry* for a set X of points if $\alpha(X) = X$.

Now consider an equilateral triangle (Figure 3.6) that has the following six symmetries:

Rotational symmetries

ρ_{120} : 120° rotation counterclockwise around its center o .

ρ_{240} : 240° rotation counterclockwise around its center o .

Reflective symmetries

σ_1 : reflection in the line L_1 .

σ_2 : reflection in the line L_2 .

σ_3 : reflection in the line L_3 .

Identity

ι : identity transformation.

Let us denote this set of symmetries of an equilateral triangle by the symbol S ; that is, $S = \{\iota, \rho_{120}, \rho_{240}, \sigma_1, \sigma_2, \sigma_3\}$. It may not be difficult to see that (S, \circ) forms an algebraic system, where \circ denotes the product of transformations. The table below expresses this composition law:

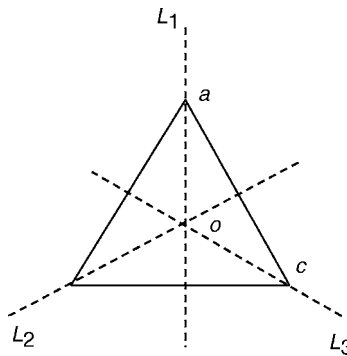


Figure 3.6 Six symmetries of an equilateral triangle $\triangle abc$ whose center is the point o

\circ	ι	ρ_{120}	ρ_{240}	σ_1	σ_2	σ_3
ι	ι	ρ_{120}	ρ_{240}	σ_1	σ_2	σ_3
ρ_{120}	ρ_{120}	ρ_{240}	ι	σ_3	σ_1	σ_2
ρ_{240}	ρ_{240}	ι	ρ_{120}	σ_2	σ_3	σ_1
σ_1	σ_1	σ_2	σ_3	ι	ρ_{120}	ρ_{240}
σ_2	σ_2	σ_3	σ_1	ρ_{240}	ι	ρ_{120}
σ_3	σ_3	σ_1	σ_2	ρ_{120}	ρ_{240}	ι

Note that the composition law is not commutative, since $\rho_{120} \circ \sigma_1 \neq \sigma_1 \circ \rho_{120}$, and so on.

We shall now show that the system (S, \circ) is isomorphic to an algebraic system of *permutation*. (For the general concept of permutation, see Example 2.44 in Section 2.5.5). For the permutation system, consider a set of three elements, say $X = \{x_1, x_2, x_3\}$. There exist six one-to-one mappings of X onto itself, which are called “permutations” of X . They can be expressed as follows:

$$\begin{array}{llllll}
 x_1 \rightarrow x_1 & x_1 \rightarrow x_1 & x_1 \rightarrow x_2 & x_1 \rightarrow x_2 & x_1 \rightarrow x_3 & x_1 \rightarrow x_3 \\
 x_2 \rightarrow x_2 & x_2 \rightarrow x_3 & x_2 \rightarrow x_1 & x_2 \rightarrow x_3 & x_2 \rightarrow x_1 & x_2 \rightarrow x_2 \\
 x_3 \rightarrow x_3 & x_3 \rightarrow x_2 & x_3 \rightarrow x_3 & x_3 \rightarrow x_1 & x_3 \rightarrow x_2 & x_3 \rightarrow x_1 \\
 p : & q : & r : & s : & t : & u :
 \end{array}$$

Let us denote these functions as p, q, r, s, t and u . Let the set of permutations be called P ; that is, $P = \{p, q, r, s, t, u\}$. Under composition of functions, say $*$, these six permutations form an algebraic system – that is, $(P, *)$ – with the following composition table:

$*$	p	q	r	s	t	u
p	p	q	r	s	t	u
q	q	p	t	u	r	s
r	r	s	p	q	u	t
s	s	r	u	t	p	q
t	t	u	q	p	s	r
u	u	t	s	r	q	p

The systems (S, \circ) and $(P, *)$ are isomorphic, since we can assume a one-to-one onto function $f : S \rightarrow P$ such that

$$\iota \leftrightarrow p; \quad \rho_{120} \leftrightarrow t; \quad \rho_{240} \leftrightarrow s; \quad \sigma_1 \leftrightarrow q; \quad \sigma_2 \leftrightarrow u; \quad \sigma_3 \leftrightarrow r;$$

which preserves the composition laws. □

Example 3.11. Consider again the algebraic system (S, \circ) , as described in Example 3.10, and the system (B, \cdot) , where $B = \{+1, -1\}$ and \cdot denotes ordinary multiplication. Consider a function $f : S \rightarrow B$ such that

$$\begin{aligned}
 f(\iota) &= 1; & f(\rho_{120}) &= 1; & f(\rho_{240}) &= 1; \\
 f(\sigma_1) &= -1; & f(\sigma_2) &= -1; & f(\sigma_3) &= -1.
 \end{aligned}$$

It is easy to see that f is a homomorphism, but not an isomorphism. In fact, it is an example of epimorphism.

Let us take another algebraic system $(C, +)$, where $C = \{\text{even}, \text{odd}\}$ and $+$ denotes addition of even and odd integers. Obviously, (B, \cdot) and $(C, +)$ are isomorphic. Thereby, $(C, +)$ is homomorphic to (S, \circ) . If we consider the mapping $f : S \rightarrow B$ such that

$$\begin{aligned} f(\iota) &= \text{even}; & f(\rho_{120}) &= \text{even}; & f(\rho_{240}) &= \text{even}; \\ f(\sigma_1) &= \text{odd}; & f(\sigma_2) &= \text{odd}; & f(\sigma_3) &= \text{odd}, \end{aligned}$$

this justifies why identify and rotation can be called *even* symmetries while reflection is called *odd* symmetry (see also Example 3.9). \square

One special case of homomorphism can be mentioned at this point. It is perfectly possible that we can consider a map of an algebraic system *into itself* that preserves the composition law. We then use the term “endomorphism.” Thus we can formalize the concept of endomorphism in the following way.

Definition 3.4: Let $(A, *)$ and $(A', *')$ be two algebraic systems such that $A' \subseteq A$. A homomorphism f from $(A, *)$ to $(A', *')$ in such a case is called an *endomorphism*. If $A' = A$, then an isomorphism from $(A, *)$ to $(A', *')$ is called an *automorphism*. \square

Example 3.12. Consider two algebraic systems $(\mathbb{Z}, +)$ and $(E, +)$, where \mathbb{Z} is the set of all integers, E is the set of even integers, and $+$ denotes addition. The function f , where $f(x) = 2x$, is an endomorphism from $(\mathbb{Z}, +)$ to $(E, +)$, since $f(x_1 + x_2) = 2x_1 + 2x_2 = f(x_1) + f(x_2)$. However, it is not an automorphism.

Let us take another example. Let $\mathbb{Z}[\sqrt{3}]$ denote the set of all numbers of the form $z = a + b\sqrt{3}$, where $a, b \in \mathbb{Z}$. Obviously, $(\mathbb{Z}[\sqrt{3}], +, \cdot)$ is an algebraic system, where $+$ and \cdot denote ordinary addition and multiplication. Now take a one-to-one onto function $\mathbb{Z} \rightarrow \mathbb{Z}$ such that $f(z) = f(a + b\sqrt{3}) = a - b\sqrt{3} = z'$, where z' belongs to \mathbb{Z} .

A little calculation shows that

$$\begin{aligned} f(z_1 + z_2) &= f(z_1) + f(z_2), \\ f(z_1 \cdot z_2) &= f(z_1) \cdot f(z_2). \end{aligned}$$

Therefore, f is an automorphism of $(\mathbb{Z}[\sqrt{3}], +, \cdot)$.

Note: The only other automorphism of this system is the *identity automorphism*, which is $a + b\sqrt{3} \leftrightarrow a + b\sqrt{3}$. \square

Example 3.13. Let us consider two algebraic systems related to geometric shapes; namely, $(\mathcal{P}(\mathbf{R}^3), \cup)$ and $(\mathcal{P}(\mathbf{R}^3), \cap)$. These two systems are automorphic under the mapping $f(A) = A^c$, where A denotes any subset of \mathbf{R}^3 ; that is, $A \in \mathcal{P}(\mathbf{R}^3)$. Obviously, f is one-to-one onto. If $A, B \in \mathcal{P}(\mathbf{R}^3)$, then

$$\begin{aligned} f(A \cup B) &= A^c \cap B^c = f(A) \cap f(B), \\ f(A \cap B) &= A^c \cup B^c = f(A) \cup f(B). \end{aligned}$$

This result explicitly demonstrates the *self-duality* of the Boolean algebra of set. However, note one important point. Given the way in which isomorphism has been defined, isomorphism

preserves the algebraic structure, but does not guarantee to preserve other kinds of structures. For example, in the present case the topological structures of the system may not be preserved, since if A is bounded, A^c becomes unbounded. \square

What is the real significance of homomorphisms? Homomorphism, particularly isomorphism, is a great simplifying device. It is important in any problem solving to be able to recognize when two apparently distinct problems are basically the same. If isomorphic structures occur in the two problems, this may give a hint about the connections between them.

We can appreciate the importance of isomorphism better from the following result.

Theorem 3.3. *Isomorphism is an equivalence relation in the collection of all algebraic systems; that is, if $(A, *)$ and $(A', *)'$ are isomorphic, we can write $(A, *) \equiv (A', *)'$.*

The proof of the theorem follows from the next two results.

Lemma 3.4. (a) *The inverse of an isomorphism is also an isomorphism. This means that if f provides an isomorphism from $(A, *)$ to $(A', *)'$, then f^{-1} gives the isomorphism from $(A', *)'$ to $(A, *)$.*

(b) *The composition of two isomorphisms is again an isomorphism. This means that if, in addition to f , g provides an isomorphism from $(A', *)'$ to $(A'', *)''$, then the composition $g \circ f$ is an isomorphism from $(A, *)$ to $(A'', *)''$.*

Thus all of the algebraic properties possessed by an algebraic system are also possessed by every other isomorphic image of the system. If $(A, *)$ is commutative, then $(A', *)'$ will be also commutative; if $(A, *)$ has a unit element, then $(A', *)'$ will also have a unit element; and so on.

3.4 Semigroups and Monoids: Two Simple Algebraic Systems

In Section 3.2, we mentioned that the structure of an algebraic system is decided by the properties possessed by its composition laws. In that section, we also defined the fundamental properties of interest. It is now possible to construct a special algebraic system from a general one by assuming that its composition laws obey some of those fundamental properties. The more properties we assume, the more structure is endowed on the system. This special algebraic system is an *abstract algebraic system* in the sense that the properties that we assume are taken as the *axioms* of the system – no other properties are assumed. If a problem in the real world, with a suitable interpretation, is found to obey the same set of axioms, we say that it is one *realization* of that abstract system. (Mathematicians call this a *model* of the abstract system.) All the structures that the abstract system possessed will be also possessed by that physical problem.

In this section, we study a few such abstract algebraic systems that we feel are relevant for the purpose of shape description. This study, however, will by no means be comprehensive. The primary purpose is to show the importance of studying abstract algebraic systems to deal the intricacies of shape description.

One of the simplest algebraic systems that we can envisage is an algebraic system that has a single composition law and is associative. Such a system is called a *semigroup*.

Definition 3.5: An algebraic system $(S, *)$ is called a *semigroup* if the composition law $*$ is associative. That is,

$$(a * b) * c = a * (b * c), \quad \text{for all } a, b, c \in S \quad (\text{associativity}).$$

□

It is more interesting to go a step further and to add one more structure to a semigroup. This gives rise to the concept of the *monoid*.

Definition 3.6: A semigroup $(M, *)$ with an identity element with respect to the composition law $*$ is called a *monoid*. That is, for all $a, b, c \in M$,

$$(a * b) * c = a * (b * c) \quad (\text{associativity}).$$

and there exists an element $e \in M$ such that, for all $a \in M$,

$$e * a = a = a * e \quad (\text{identity}).$$

□

Example 3.14. We know that $(\mathcal{P}(\mathbf{R}^3), \cup)$ is an algebraic system and that \cup is associative. Therefore, it is a semigroup. In fact, it is an *Abelian semigroup* (named after the Norwegian mathematician Niels Henrik Abel), since the composition law is also commutative. Furthermore, it has an identity element, which is the empty set \emptyset . Thus, $(\mathcal{P}(\mathbf{R}^3), \cup)$ is a monoid. Similarly, $(\mathcal{P}(\mathbf{R}^3), \cap)$ and (K, \cap) (see Example 3.2) are also monoids, with identity element \mathbf{R}^3 in both cases. For the purpose of shape description, this is an important characterization. □

Example 3.15. Semigroups and monoids have special significance in string processing and language theory. The following example may give you some insight. Let $V = \{a, b, c\}$ be a nonempty set of symbols that is *finite*. (In language theory, such a set is generally called an *alphabet*.) We will now define V^+ as the set of all *strings* of symbols taken from V . So V^+ will include $a, b, c, aa, ab, ba, aabc, abbcc, \dots$, and so on. (To continue the linguistic analogy, strings are also called *words*.) V^+ is infinite.

On V^+ , we can define a composition law \odot , commonly known as the *concatenation* operation, such that if $\alpha, \beta \in V^+$, then $\alpha \odot \beta = \alpha\beta$; that is, we can write down the string α and follow it immediately with the string β . Thus $aa \odot bc = aabc$, $abb \odot cc = abbcc$, and so on. Clearly, (V^+, \odot) forms an algebraic system. It is also easy to see that the concatenation \odot is associative. Therefore, (V^+, \odot) is a semigroup, and called a *free semigroup*, generated by the alphabet V .

Let us now add to the set V^+ a special string that is invisible; in other words, empty. Let us use a special symbol for an empty string – say, Λ . Then, for all $\alpha \in V^+$, $\alpha \odot \Lambda = \alpha = \Lambda \odot \alpha$. Let $V^* = \{V^+, \Lambda\}$. The algebraic system (V^*, \odot) is a monoid and Λ is the identity with respect to \odot .

For practical purposes, the entire set V^* may not be of much use, but just a subset of it. A *language* (over the alphabet V) is a subset of the set V^* . □

Example 3.16. Let \mathbb{N} be the set of natural numbers; that is, $\mathbb{N} = \{0, 1, \dots\}$. Then $(\mathbb{N}, +)$ and (\mathbb{N}, \cdot) are monoids with the identities 0 and 1, respectively. On the other hand, if E^+ denotes the set of positive even numbers, then $(E^+, +)$ and (E^+, \cdot) are semigroups, but not monoids. \square

In the previous sections, we have introduced two general concepts for an algebraic system; namely, the concept of subsystems and the concept of morphisms. Naturally, these two concepts are applicable for semigroups and monoids. For example, a *monoid homomorphism* can be defined as follows.

Definition 3.7: Let $(M_1, *)$ and (M_2, Δ) be any two monoids. A function $f : M_1 \rightarrow M_2$ such that for any two elements $a, b \in M_1$,

$$f(a * b) = f(a) \Delta f(b) \quad \text{and} \quad f(e_{M_1}) = e_{M_2}$$

is called a *monoid homomorphism*. The symbols e_{M_1} and e_{M_2} denote the identity elements in $(M_1, *)$ and (M_2, Δ) , respectively. (In the case where f is an onto mapping, the first condition implies the second.) \square

In fact, Example 3.13 can be considered to be an example of semigroup automorphism. It is easy to show that *semigroup homomorphism* preserves the associativity property, as well as the commutativity property for an Abelian semigroup. Similar results are also true for *monoid homomorphism*.

3.5 Groups

3.5.1 Fundamentals

The real fun and usefulness of algebraic systems begins when we go one step further and add another structure to the monoid. This structure – that is, a further restriction imposed on the elements of the monoids, namely, the existence of an inverse for each element – results in an algebraic system called a *group*. (The technical term “group” was first used by a French Republican called Evariste Galois (1811–1832).) The theory of groups was conceived about 100 years ago to aid in solving for the roots of a polynomial. Since then, group theory has been widely applied in various branches of the physical sciences, in computer science, and particularly in geometry. Group theory is very extensive, but we touch on only a few points here.

Definition 3.8: A *group* $(G, *)$ is an algebraic system in which the composition law $*$ on the set G satisfies three conditions:

1. For all $a, b, c \in G$,

$$(a * b) * c = a * (b * c) \quad (\text{associativity}).$$

2. There exists an element $e \in G$ such that, for all $a \in G$,

$$e * a = a = a * e \quad (\text{identity}).$$

3. For every $a \in G$, there exists an element, denoted by a^{-1} , in G such that

$$a * a^{-1} = e = a^{-1} * a \quad (\text{inverse}).$$

□

These three conditions are also called the *defining axioms of a group*.

Note: We recall that the identity element e and inverse a^{-1} of any element a must be unique (see Section 3.2).

If G is a group with the additional property that composition law $*$ is commutative, we call G a *commutative group*, or more frequently an *Abelian group*.

We will now give a few preliminary examples of groups.

Example 3.17. It may not be difficult for you to identify the fact that the algebraic system $(\mathbb{R}, +)$ is an Abelian group, and so is $(\mathbb{Z}, +)$. On the other hand, the algebraic system (\mathbb{R}, \cdot) is not a group, since the element $0 \in \mathbb{R}$ does not have its inverse. Thus only if we exclude 0 from \mathbb{R} , the system $(\mathbb{R} - \{0\}, \cdot)$ becomes an Abelian group. Also, the systems $(E, +)$, $(\mathbb{C}, +)$ are well-known Abelian groups. □

Example 3.18. Let $G = \{e, a, b, c\}$ be a set of four elements with the following composition table:

$*$	e	a	b	c
e	e	a	b	c
a	a	b	c	e
b	b	c	e	a
c	c	e	a	b

The algebraic system $(G, *)$ is an Abelian group.

With the same set G , we can envisage another composition law – say, \circ – that can be expressed as follows:

\circ	e	a	b	c
e	e	a	b	c
a	a	e	c	b
b	b	c	e	a
c	c	b	a	e

The system (G, \circ) is also an Abelian group. □

Note: The interesting fact is that, with four elements in G , these two are the *only* possible structures that can form groups.

Example 3.19. We mentioned in Example 3.14 that the algebraic systems $(\mathcal{P}(\mathbb{R}^3), \cup)$ and $(\mathcal{P}(\mathbb{R}^3), \cap)$ are monoids, but they are not groups because it is not possible to define inverse elements in those systems. The question is: “Is it possible to define any algebraic system in $\mathcal{P}(\mathbb{R}^3)$ that is a group?”

Let us define on $\mathcal{P}(\mathbf{R}^3)$ the composition law Δ , called the *symmetric difference*, which is given by (see Section 2.3.6)

$$\begin{aligned} A \Delta B &= (A \cup B) - (A \cap B), & \text{where } A, B \in \mathcal{P}(\mathbf{R}^3) \\ &= (A - B) \cup (B - A). \end{aligned}$$

It was shown that the composition law Δ was associative. The identity element is the empty set \emptyset , since $A \Delta \emptyset = A = \emptyset \Delta A$. Since $A \Delta A = \emptyset$, it is clear that every element A has an inverse that is itself. Thus $(\mathcal{P}(\mathbf{R}^3), \Delta)$ is a group, and in fact it is an Abelian group. (It appears to us that it might be an interesting idea to design a shape description scheme with Δ as an operator.) \square

Example 3.20. Consider the set A , which consists of the identity transformation ι and only one reflection, say σ , on the plane; that is, $A = \{\iota, \sigma\}$, and the algebraic system (A, \circ) as described in Example 3.9. Since $\sigma \circ \sigma = \iota$, the element σ has an inverse that is itself. Thus (A, \circ) is a group. \square

When G set forms a group, the number of elements in the set G is an important factor in understanding the nature of that group. The note in Example 3.18 may be illuminating in this respect. This number is called the *order* of a group. Its formal definition is as follows.

Definition 3.9: The *order* of a group $(G, *)$, normally denoted by $|G|$, is the number of elements of G , when G is finite. \square

Thus the order of the group described in Example 3.18 is four, while that in Example 3.20 is two. If G contains a finite number of elements, we say that $(G, *)$ is a *finite group*.

Let us give a few more examples of groups and their orders that will prove useful for our later discussion.

Example 3.21. (permutation group). You may recall (Section 2.5.5) that any one-to-one mapping of a finite set X onto itself is called a *permutation* of X . For instance, the set $X = \{1, 2, 3, 4, 5\}$, which consists of five digits, has a permutation – say, p_1 – such that

$$p_1(1) = 2, \quad p_1(2) = 3, \quad p_1(3) = 4, \quad p_1(4) = 5, \quad p_1(5) = 1.$$

Another permutation might be p_2 , given by

$$p_2(1) = 2, \quad p_2(2) = 3, \quad p_2(3) = 1, \quad p_2(4) = 5, \quad p_2(5) = 4,$$

and so on.

Since permutations are simply special functions, we can consider the composition of permutations. For example, in this case the composition of p_1 and p_2 – that is, $p_1 \circ p_2$ – works out as follows:

$$\begin{aligned} p_1 \circ p_2(1) &= p_1(p_2(1)) = p_1(2) = 3, & p_1 \circ p_2(2) &= 4, & p_1 \circ p_2(3) &= 2, \\ p_1 \circ p_2(4) &= 1, & p_1 \circ p_2(5) &= 5. \end{aligned}$$

Note that $p_1 \circ p_2$ is another permutation of X . Furthermore, $p_1 \circ p_2 \neq p_2 \circ p_1$. Thus we can generalize this observation into the following theorem.

Theorem 3.5. *Let the set of all permutations of a set X be denoted by S_X ; or, in special case in which X has n elements, by S_n . Then:*

- (a) *The algebraic system (S_X, \circ) , where \circ denotes composition of permutation, forms a group. This group is called the permutation group.*
- (b) *If the set X has n elements, then the order of the group (S_X, \circ) is $n!$.*
- (c) *The group is Abelian if and only if X has one element or two elements.*

Proof: (a) The identity transformation ι is obviously a permutation of X . If p_1 and p_2 are permutations of X , then $p_1 \circ p_2$ is also a permutation of X . Since any permutation p_1 is a one-to-one onto function, its inverse p_1^{-1} is also one-to-one onto and it belongs to S_X ; that is, p_1^{-1} is also a permutation of X . The composition of functions are associative. Thus (S_X, \circ) is a group.

(b) Let us order the n elements of X as x_1, x_2, \dots, x_n . If we are to construct a one-to-one function f from X into X , then we have n choices for $f(x_1)$, $(n - 1)$ choices for $f(x_2)$, \dots , and 1 choice for $f(x_n)$. Thus there are $n(n - 1) \cdots (1) = n!$ possible one-to-one functions from X into X . Since X is finite, each of these one-to-one functions is also onto. Hence there are $n!$ permutations of X .

(c) This part is fairly obvious. □

In representing permutations, a nice notational system can be thought of by utilizing the concept of the *cycle*. Consider, for example, the permutation p_2 above, which can be written as

$$p_2 : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 5 & 4 \end{pmatrix}.$$

Starting with the symbol 1, we see that the permutation takes 1 to 2, 2 to 3, and 3 to 1, closing a cycle, which we can write as (123). Continuing, we find that there is a cycle (45). So the permutation p_2 can alternatively be expressed as

$$p_2 = (123)(45).$$

Similarly,

$$p_1 = (12345).$$

This cycle notation can, in fact, be used for expressing the permutation of any set – not necessarily sets like $\{1, 2, 3, \dots\}$. This is because if a set X is given as $\{a, b, c, \dots\}$, we can always level its elements as 1, 2, 3, \dots .

A cycle containing two symbols (a 2-cycle) is called a *transposition*. Note that a transposition is the most primitive cycle, and it can be used as a building block. Any cycle can be written as

a product of transpositions (that have elements in common). Thus

$$(123) = (13)(12),$$

and, in general,

$$(12 \dots n) = (1n) \dots (13)(12).$$

□

If a permutation is the composition of an even number of transpositions, it is called the *even permutation*. If the number is odd, it is called the *odd permutation*.

An n -cycle is an even permutation if n is odd, and an odd permutation if n is even! An n -even permutation is denoted by A_n .

Theorem 3.6. *The number of members of the n -even permutation A_n is given as [33],*

$$|A_n| = \frac{1}{2} n! = \frac{1}{2} |S_n|, \quad (3.1)$$

where S_n is the permutation group on a set having n elements (see Theorem 3.5).

Example 3.22 (a group and its practical models). Let $G = \{e, a, b, c, d, f\}$ be a set of six elements with the composition rule

*	e	a	b	c	d	f
e	e	a	b	c	d	f
a	a	b	e	d	f	c
b	b	e	a	f	c	d
c	c	f	d	e	b	a
d	d	c	f	a	e	b
f	f	d	c	b	a	e

Clearly, the algebraic system $(G, *)$ is a group, but it is not Abelian.

We can envisage several instances of this group, two of which are as follows:

1. First, we present a simple instance from arithmetic. Start with any number that is different from 0 and 1 – say, x . Let us generate all the new numbers from x by two operations: (i) subtracting x from 1 and (ii) dividing 1 by x . In the first pass, we shall generate two new numbers $1 - x$ and $\frac{1}{x}$. Repeat these two operations on the new numbers. Then $1 - x$ gives back x and a new number $\frac{1}{1-x}$; $\frac{1}{x}$ gives the new number $1 - \frac{1}{x} = \frac{x-1}{x}$, and also x . Repeat this several times and you simply get one or another of the six numbers $x, \frac{1}{x}, 1 - x, \frac{1}{1-x}, \frac{x-1}{x}, \frac{x}{x-1}$.

Now assume that in the set G , the elements are all various transformations such that

$$\begin{aligned} e(x) &= x, & a(x) &= \frac{1}{1-x}, & b(x) &= \frac{x-1}{x}, \\ c(x) &= \frac{1}{x}, & d(x) &= 1-x, & f(x) &= \frac{x}{x-1}. \end{aligned}$$

With this interpretation of the elements of G , and assuming that $*$ denotes composition of transformations, the composition table is satisfied.

2. The second instance is by means of a permutation group. Let the set $X = \{1, 2, 3\}$ and let all six possible permutations of its elements be

$$p_1 : \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \iota, \quad p_2 : \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = (123),$$

$$p_3 : \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} = (132), \quad p_4 : \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = (23),$$

$$p_5 : \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} = (13), \quad p_6 : \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = (12).$$

Write

$$p_1 = e, \quad p_2 = a, \quad p_3 = b, \quad p_4 = c, \quad p_5 = d, \quad p_6 = f.$$

Now it is easy to see that under composition of permutations, the above composition table is again satisfied. \square

In group theory, we frequently come across a particular notation. Because of the associative law, in any group the symbol a^k is perfectly well defined as a shorthand for

$$a^k \equiv a * a * \cdots * a,$$

with k terms. It is advantageous to introduce the notation

$$a^0 = e, \quad \text{and also} \quad a^{-k} \equiv a^{-1} * a^{-1} * \cdots * a^{-1},$$

with k terms. Such a notation can be termed as a *power* notation.

It can be verified that with these conventions, most of our usual concepts of “power” can be extended for a group too. For example,

$$a^n * a^m = a^m * a^n = a^{n+m},$$

for any $n, m = 0, \pm 1, \pm 2, \dots$ Moreover,

$$(a^n)^m = (a^m)^n = a^{nm}, \quad \text{and also} \quad a^{-n} = (a^n)^{-1}.$$

However, note that if $a \neq b$ and the group is not Abelian, then, in general,

$$(a * b)^n \neq a^n * b^n.$$

3.5.2 The Advantages of Identifying a System as a Group

The usefulness of identifying an algebraic system as a group will become more apparent as we go deeper into our discussions on groups in the following subsections. However, to motivate you to enter into those discussions, we briefly mention, at this stage, some elementary advantages of groups.

- (a) If an algebraic system forms a group (particularly an Abelian group), the composition and manipulation of its elements can be carried out just like the traditional methods of algebra. For example, within a group $(G, *)$, we can solve the equation

$$a * x = b,$$

where $a, b \in G$, as

$$x = a^{-1} * b.$$

(However, if the group is not Abelian, then the solution of equation $x * a = b$ will be different, since the solution $x = b * a^{-1}$ may not be equal to $a^{-1} * b$.) One of the greatest advantages in a group is the applicability of the *cancellation law* as we apply it in traditional algebra; that is, $a * c = b * c$ implies $a = b$ and, similarly, $c * a = c * b$ also implies $a = b$.

- (b) From the “synthesis” as well as from the “analysis” point of view, one primary advantage is that a proper subset of the set G may generate the whole set G . For example, consider the group $(\mathbb{Z}, +)$. Take the subset $Z' = \{0, 1\}$. Now, repeated application of $+$ on the element 1 will generate the numbers 1, 2, 3, \dots . Since $(\mathbb{Z}, +)$ is a group, all these numbers must belong to \mathbb{Z} . Moreover, the inverse of these numbers – that is, $-1, -2, -3, \dots$ – must also be included in \mathbb{Z} . Thus $Z' \subset \mathbb{Z}$ can generate the whole set \mathbb{Z} . For this reason, such a subset is called the set of *generators* for the group. It becomes more interesting if a single element, say, $a \in G$ can generate the whole set G . In other words, every element of G can be written as some power of a ; that is, a^k for some integer k . Such a group $(G, *)$ is said to be *cyclic*. If G is a finite cyclic group, we can express G as follows:

$$G = \{a, a^2, a^3, \dots, a^n = e\},$$

where n is the least positive integer for which $a^n = e$. Obviously, the order of the group is n . Consider group $(G, *)$ in Example 3.18. It is a cyclic group of order 4, since $a^2 = b, a^3 = a * b = c, a^4 = b * b = e$. In geometry, we frequently come across such cyclic groups. We present one such example here.

Example 3.23 (cyclic group C_n of rotations). Consider a two-dimensional object F on the plane that is being rotated around some point o through an angle $\theta = \frac{2\pi}{n}$ (n is an integer). See Figure 3.7. If we rotate the figure F by angles $\theta, 2\theta, 3\theta, \dots$, then finally the figure must be brought into coincidence with itself when rotated through an angle $n\theta$. Thus in the plane the set of successive rotations $\{\rho_{\frac{2\pi}{n}}, \rho_{\frac{2 \cdot 2\pi}{n}}, \dots, \rho_{\frac{n \cdot 2\pi}{n}}\}$ forms a group under composition of transformations, and obviously a cyclic group. To see this, denote $\rho_{\frac{2\pi}{n}} = \rho_\theta$, and observe that $\rho_{\frac{k \cdot 2\pi}{n}} = \rho_\theta \circ \rho_\theta \circ \dots \circ k \text{ times} = \rho_\theta^k$. This cyclic group is generally denoted by a special symbol C_n in geometry. The pivotal point is sometimes called the *center of symmetry*. The order of C_n is clearly n . \square

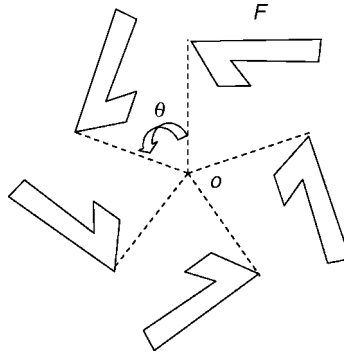


Figure 3.7 An example of a cyclic group C_n of rotations

- (c) If we encounter an algebraic system that forms a group, even from the order of the group it is possible to deduce various properties of the system. Let us give some examples. A group of order 1 has only the identity element; that is, it is the group $(\{e\}, *)$. A group of order 2 is always the group $(\{e, a\}, *)$, where a is an element other than the identity element e and $a * a = e$; that is, $a = a^{-1}$ (in the case of geometric transformations, a is called involution). It is easy to see that all of the groups of order 2 are isomorphic; that is, essentially the same. In other words, all of the groups of order 2 are isomorphic to the group (A, \circ) , as described in Example 3.20. In the same way, it can be shown that all groups of order 3 – that is, $(\{e, a, b\}, *)$ – must be isomorphic to the group whose composition rule is given below:

$*$	e	a	b
e	e	a	b
a	a	b	e
b	b	e	a

The groups of order 4 are isomorphic to one of the groups $(G, *)$ or (G, \circ) , as given in Example 3.18.

It is also not difficult to show that all groups up to order 5 are Abelian.

- (d) The concept of groups becomes particularly interesting when we explore the properties of various geometric transformations. This particular aspect of groups will be discussed in Sections 3.7 and 3.8. Once a set of geometric transformations can be identified as forming a group, their algebraic realization becomes a trivial step. The algebraic realization, in turn, allows us to manipulate the transformations mechanically, just as we manipulate symbols in ordinary algebra.

3.5.3 Transformation Groups

Before going into other concepts in the theory of groups, we have decided to talk about a special group, called the *transformation group*, which is particularly important in dealing with geometric objects as an action on a set. Formally, it is defined in the following way.

Definition 3.10: Let X be an arbitrary set. Let $T = \{t_1, \dots, t_i, \dots\}$ be some set of one-to-one onto functions

$$t_i : X \rightarrow X$$

subject to the following criteria:

- (i) the identity function 1_X belongs to T ;
- (ii) if $t_a \in T$, then $t_a^{-1} \in T$;
- (iii) if t_a and t_b belong to T , then the composition of these two functions – that is, $t_a \circ t_b$ – also belongs to T .

The algebraic system (T, \circ) is a group under composition of functions, and is called a *transformation group* on the set X . \square

For notational convenience, $t_a \circ t_b$ is frequently expressed as $t_a t_b$ and is called the *product* of two functions instead of the composition of two functions.

Example 3.24 (the group of symmetries of an equilateral triangle). We have already given several examples of transformation groups in our earlier discussions. Consider Example 3.10. There we mentioned that if S is the set of symmetries of an equilateral triangle – that is, $S = \{\iota, \rho_{120}, \rho_{240}, \sigma_1, \sigma_2, \sigma_3\}$ – then (S, \circ) forms an algebraic system, where \circ denotes the product of transformations. By examining the composition table of (S, \circ) , it becomes immediately apparent that it is a group – though not an Abelian group. Therefore, (S, \circ) is a transformation group on a set of points X forming an equilateral triangle.

It might be of interest to note that a subset S_1 of S forms another smaller group, where $S_1 = \{\iota, \rho_{120}, \rho_{240}\}$, since $\rho_{120} \circ \rho_{120} = \rho_{120}^2 = \rho_{240}$, $\rho_{120}^3 = \rho_{120} \circ \rho_{240} = \rho_{240} \circ \rho_{120} = \iota$, and so on. In fact, (S_1, \circ) is a *cyclic group* of order 3 because $S_1 = \{\rho_{120}, \rho_{120}^2, \rho_{120}^3 = \iota\}$.

Similarly, $S_2 = \{\iota, \sigma_1\} \subset S$ also forms a smaller group, since $\sigma_1^2 = \iota$, $\sigma_1^{-1} = \sigma_1$. This is also a cyclic group and its order is 2. In the same way, $S_3 = \{\iota, \sigma_2\}$ and $S_4 = \{\iota, \sigma_3\}$ can form separate cyclic groups.

It must be evident that (S_1, \circ) , (S_2, \circ) , (S_3, \circ) , (S_4, \circ) are all *subsystems* of the algebraic system (S, \circ) . Since (S, \circ) is a group, we call all these subsystems the *subgroups* of (S, \circ) . \square

Example 3.25 (the group of symmetries of a square). Consider the symmetries of a square on the plane (Figure 3.8).

It is clear that the square has eight symmetries.

Rotational symmetries

ρ_{90} : 90° rotation counterclockwise around its center o .

ρ_{180}, ρ_{270} : similar rotations by 180° and 270°, respectively.

Reflective symmetries

σ_1 : reflection in the vertical line A_4 .

σ_2 : reflection in the horizontal line A_2 .

σ_3 : reflection in the diagonal line A_1 .

σ_4 : reflection in the diagonal line A_3 .

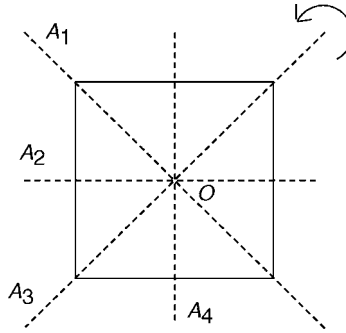


Figure 3.8 The eight symmetries of a square on the plane

Identity

ι : identity transformation.

Let us denote this set of symmetries of a square by S ; that is,

$$S = \{\iota, \rho_{90}, \rho_{180}, \rho_{270}, \sigma_1, \sigma_2, \sigma_3, \sigma_4\}.$$

Again it can be easily verified that (S, \circ) is a group (not Abelian), where \circ denotes composition of transformations.

This is a transformation group of the set X forming a square.

We again note that $(\{\iota, \rho_{90}, \rho_{180}, \rho_{270}\}, \circ)$, $(\{\iota, \sigma_1\}, \circ)$, and so on are subgroups of (S, \circ) . \square

3.6 Symmetry Groups

3.6.1 The Action of a Group on a Set

The concept of a group acting on a set X is a slight generalization of the group of permutations of X . It is equivalent to considering a subgroup of the permutation group, which was defined in Examples 2.21 and 3.22). This concept is useful for determining the order of the permutation groups of solids in three dimensions.

The group (G, \cdot) acts on the set X if there is a function

$$\psi : G \times X \rightarrow X$$

such that when we write $g(x)$ for $\psi(g, x)$, we have:

1. $(g_1 g_2)(x) = g_1(g_2(x))$, for all $g_1, g_2 \in G, x \in X$.
2. $e(x) = x$, if e is the identity of G and $x \in X$.

Definition 3.11 (stabilizer): If G acts on a set X and $x \in X$, then

$$\text{Stab } x = \{g \in G \mid g(x) = x\}$$

is a subgroup of G , called the *stabilizer* of x . It is the set of elements of G that fix x . \square

Definition 3.12 (orbit): The set of all images of an element $x \in X$ under the action of a group G is called the *orbit* of x under G and is denoted by

$$\text{Orb } x = \{g(x) \mid g \in G\}.$$

\square

Then, we have an important connection between the number of elements in the orbit of a point x and the stabilizer of that point.

Theorem 3.7. *If the finite group G acts on a set X , then for each $x \in X$,*

$$|G| = |\text{Stab } x| |\text{Orb } x|.$$

Example 3.26. Find the number of proper rotations of a cube.

Let G be the group of proper rotations of a cube; that is, rotations that can be carried out in three dimensions. The stabilizer of vertex 1 in Figure 3.9 is $\text{Stab } 1 = \{(1), (245) \circ (386), (254) \circ (368)\}$. The orbit of 1 is the set of all the vertices, because there is an element of G that will take 1 to any other vertex. Therefore, by Theorem 3.7,

$$|G| = |\text{Stab } 1| |\text{Orb } 1| = 3 \cdot 8 = 24.$$

\square

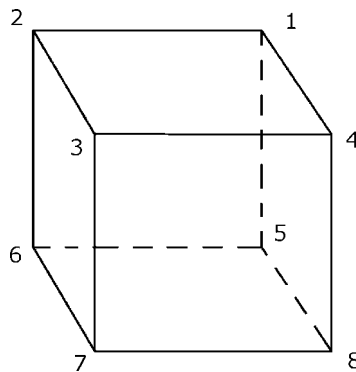


Figure 3.9 A cube under proper rotation group G

3.6.2 Translations and the Euclidean Group

Euclidean geometry in n dimensions is concerned with the bijections $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^n$ that preserve distances in Euclidean n -space (i.e., rigid motions). This group is isometric and the group of all isometries of \mathbb{R}^n is called the *Euclidean group* in n dimensions. We denote it by $E(n)$. Given $w \in \mathbb{R}^n$, the mapping of $\mathbb{R}^n \rightarrow \mathbb{R}^n$ with $v \rightarrow v + w$ is called translation by w . The group $T(n)$ of all translations is a subgroup of $E(n)$.

Recall that a function $\lambda : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is called a *linear transformation* if $\lambda(av + bw) = a\lambda(v) + b\lambda(w)$ for all $a, b \in \mathbb{R}$ and all $v, w \in \mathbb{R}^n$. Let $\{e_1, e_2, \dots, e_n\}$ denote the *standard basis* of \mathbb{R}^n ; that is, the columns of the $n \times n$ identity matrix. Then the action of λ is matrix multiplication $\lambda(v) = Av$ for all v in \mathbb{R}^n , where the matrix A is given in terms of its columns by $A = [\lambda(e_1), \lambda(e_2), \dots, \lambda(e_n)]$ and is called the *standard matrix* of α .

A function $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is said to be an *isometry* if

$$\|\alpha(v) - \alpha(w)\| = \|v - w\|, \quad \text{for all } v, w \in \mathbb{R}^n. \quad (3.2)$$

Since $\|v - w\|^2 = \|v\|^2 + 2(v \cdot w) + \|w\|^2$, it follows from (3.2) that every isometry α preserves inner products in the sense that

$$\alpha(v) \cdot \alpha(w) = v \cdot w, \quad \text{for all } v, w \in \mathbb{R}^n. \quad (3.3)$$

Lemma 3.8. *If $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is an isometry such that $\alpha(0) = 0$, then α is linear.*

Proof: It follows from (3.3) that $\{\alpha(e_1), \alpha(e_2), \dots, \alpha(e_n)\}$ is an orthonormal basis of \mathbb{R}^n . If $a \in \mathbb{R}$ and $v \in \mathbb{R}^n$, then (3.3) implies that

$$[\alpha(av) - a\alpha(v)] \cdot \alpha(e_i) = (av) \cdot e_i - a(v \cdot e_i) = 0, \quad \text{for each } i.$$

Hence $\alpha(av) = a\alpha(v)$, and $\alpha(v + w) = \alpha(v) + \alpha(w)$ follows in the same way for all $v, w \in \mathbb{R}^n$. \square

Hence the isometries of \mathbb{R}^n that fix the origin are precisely the linear isometries. An $n \times n$ matrix A is called *orthogonal* if it is invertible and $A^{-1} = A^\top$; and, equivalently, if the columns of A are an orthonormal basis of \mathbb{R}^n . These matrices form a subgroup of the group of all invertible matrices, called the *orthogonal group* and denoted by $O(n)$.

Proposition 3.9. *Let $\lambda : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a linear transformation with matrix A . Then:*

- (1) λ is an isometry if and only if A is an orthogonal matrix.
- (2) The group of linear isometries of \mathbb{R}^n is isomorphic to $O(n)$.

Proof: If A is orthogonal, then for all $v, w \in \mathbb{R}^n$,

$$\begin{aligned} \|\lambda(v) - \lambda(w)\|^2 &= A(v - w) \cdot A(v - w) \\ &= (v - w)^\top A^\top A(v - w) = \|v - w\|^2 \end{aligned}$$

and it follows that λ is an isometry. Conversely, if λ is an isometry and $\{e_1, e_2, \dots, e_n\}$ is the standard basis of \mathbb{R}^n , then (3.3) gives

$$\begin{aligned}
e_i \cdot e_j &= \lambda(e_i) \cdot \lambda(e_j) = Ae_i \cdot Ae_j \\
&= e_i^\top (A^\top A) e_j = \text{the } (i, j)\text{-entry of } A,
\end{aligned}$$

for all i and j . It follows that $A^\top A = I$, so A is orthogonal, proving (1). But then the correspondence $\lambda \leftrightarrow A$ between the linear transformation λ and its standard matrix A induces a group isomorphism between the (linear) isometries, fixing the origin and the orthogonal matrices. This proves (2). \square

3.6.3 The Matrix Group

A multiplicative group whose elements are $n \times n$ complex matrices is called a *matrix group* if its identity element is the $n \times n$ identity matrix I . For example, if

$$A_k = \begin{bmatrix} \cos(2\pi k/m) & -\sin(2\pi k/m) \\ \sin(2\pi k/m) & \cos(2\pi k/m) \end{bmatrix},$$

then $(\{A_0, A_1, \dots, A_{m-1}\}, \cdot)$ is a real matrix group. The matrix A_k represents a counterclockwise rotation of the plane about the origin through an angle $(2\pi k/m)$.

If G is a real matrix group, the determinant of any element of G is either $+1$ or -1 . If G is a complex matrix group, the determinant of any element is of the form $e^{2\pi k/m}$.

The orthogonal group $O(n)$ is a real matrix group, and therefore any element must have determinant $+1$ or -1 . The determinant function

$$\det : O(n) \rightarrow \{1, -1\}$$

is a group morphism from $(O(n), \cdot)$ to $(\{1, -1\}, \cdot)$. The kernel, consisting of orthogonal matrices with determinant $+1$, is called the *special orthogonal group* of dimension n and is denoted by

$$SO(n) = \{A \in O(n) \mid \det A = +1\}.$$

The elements of $SO(n)$ are called *proper rotations*, whereas the elements in the other coset of $O(n)$ by $SO(n)$, consisting of orthogonal matrices with determinant -1 , are called *improper rotations*.

Proposition 3.10. (1) *The set of proper rotations in two dimensions is*

$$SO(2) = \left\{ \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \mid \theta \in \mathbb{R} \right\}.$$

(2) *The set of improper rotations in two dimensions is*

$$\left\{ \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{bmatrix} \mid \theta \in \mathbb{R} \right\}.$$

(3) *The eigenvalues of the proper rotation*

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

are $e^{\pm i\theta}$ and those of any improper rotation are ± 1 .

The improper rotation

$$B = \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{bmatrix}$$

always has an eigenvalue 1 and hence leaves an axis through the origin invariant because, for any corresponding eigenvector x , $Bx = x$. It can be verified that this axis of eigenvectors, corresponding to the eigenvalue 1, is a line through the origin making an angle $\theta/2$ with the first coordinate axis. The matrix B corresponds to a reflection of the plane about this axis.

Hence we see that an improper rotation is a reflection about a line through the origin and, conversely, it is easy to see that a reflection about a line through the origin is an improper rotation.

3.7 Proper Rotations of Regular Solids

3.7.1 The Symmetry Groups of the Regular Solids

One class of symmetries that we know occurs in three dimensions is the class of symmetry groups of the regular solids: the tetrahedron, the cube, the octahedron, the dodecahedron, and the icosahedron. In this section, we determine the *proper* rotation groups of these solids. These will all be subgroups of $SO(3)$. We restrict our consideration to proper rotations because these are the only ones that can be physically performed on models in three dimensions; to physically perform an improper symmetry on a solid, we would require four dimensions!

Theorem 3.11. *Every element $A \in SO(3)$ has a fixed axis, and A is a rotation about that axis.*

Proof: Let λ_1, λ_2 , and λ_3 be the eigenvalues of A . These are the roots of the cubic characteristic polynomial with real coefficients. Hence at least one eigenvalue is real and if a second one is complex, the third is its complex conjugate, and $|\lambda_1| = |\lambda_2| = |\lambda_3| = 1$. \square

Since $\det A = \lambda_1 \lambda_2 \lambda_3 = 1$, we can relabel the eigenvalues, if necessary, so that one of the following cases occurs:

1. $\lambda_1 = \lambda_2 = \lambda_3 = 1$.
2. $\lambda_1 = 1, \lambda_2 = \lambda_3 = -1$.
3. $\lambda_1 = 1, \lambda_2 = \bar{\lambda}_3 = e^{i\theta}$ (where $\theta \neq n\pi$).

In all cases, there is an eigenvalue equal to 1. If x is a corresponding eigenvector, then $Ax = x$, and A fixes the axis along the vector x . We can change the coordinate axes so that A can be

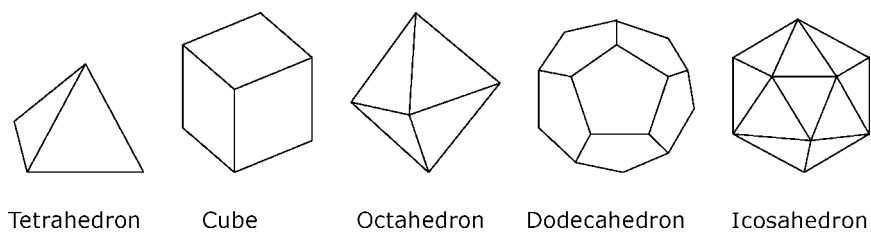


Figure 3.10 Regular solids

written in one of the following three forms:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}.$$

The first matrix is the identity, the second is a rotation through π , and the third is a rotation through θ about the fixed axis. □

A *regular solid* is a polyhedron in which all faces are congruent regular polygons and all vertices are incident with the same number of faces. There are five such solids, and they are illustrated in Figure 3.10; their structure is given in Table 3.1. Readers interested in making models of these polyhedra should consult Cundy and Rollett [15].

Given any polyhedron, we can construct its *dual* polyhedron in the following way. The vertices of the dual are the centers of the faces of the original polyhedron. Two centers are joined by an edge if the corresponding faces meet in an edge. The dual of a regular tetrahedron is another regular tetrahedron. The dual of a cube is an octahedron, and the dual of an octahedron is a cube. The dodecahedron and icosahedron are also duals of each other. Any symmetry of a polyhedron will induce a symmetry on its dual and vice versa. Hence dual polyhedra will have the same rotation group.

Theorem 3.12. *The group of proper rotations of a regular tetrahedron is isomorphic to A_4 .*

Table 3.1 Regular solids

Polyhedron	Faces	Edges	Vertices	Faces at Vertices
Tetrahedron	4 triangles	6	4	3
Cube	6 squares	12	8	3
Octahedron	8 triangles	12	6	4
Dodecahedron	12 pentagons	30	20	3
Icosahedron	20 triangles	30	12	5

(Here, A_4 means 4-even permutation, which is a permutation of the composition of four transpositions. See the definition just after Theorem 3.5 in Section 3.5.1.)

Proof: Label the vertices of the tetrahedron 1, 2, 3, and 4. Then any rotation of the tetrahedron will permute these vertices. So if G is the rotation group of the tetrahedron, we have a group morphism $f: G \rightarrow S_4$ whose kernel contains only the identity element. Hence, by the morphism theorem, G is isomorphic to Image of f . \square

We can use Theorem 3.7 to count the number of elements of G . The stabilizer of the vertex 1 is the set of elements fixing 1, and is $\{(1), (234), (243)\}$. The vertex 1 can be taken to be any of the four vertices under G , so the orbit of 1 is the set of four vertices. Hence, $|G| = |\text{Stab } 1| |\text{Orb } 1| = 3 \cdot 4 = 12$.

There are two types of nontrivial elements in G that are illustrated in Figures 3.11(a) and (b). For (a), there are rotations of order 3 about axes, each of which joins a vertex to the center of the opposite face. These rotations perform an even permutation of the vertices, because each fixes one vertex and permutes the other three cyclically. The set of permutation elements fixing the vertex 1 is $\{(1), (234), (243)\}$, for example. We have four choices of the fixing vertex. All of them are even permutations.

There are also rotations of order 2 about axes, each of which joins the midpoints of a pair of opposite edges, as shown in Figure 3.11(b). (Two edges in a tetrahedron are said to be opposite if they do not meet.) The corresponding permutations interchange two pairs of vertices and, being products of two transpositions, as $(12) \circ (34)$ for example, are even. \square

Hence, all of permutations are even, and Image of $f = A_4$.

The group A_4 is sometimes called the *tetrahedral group*.

Theorem 3.13. *The group of proper rotations of a regular octahedron and cube is isomorphic to S_4 .*

(Here, the symmetric group S_4 was also given in Theorem 3.5 as was the set of all permutations of a set that has n elements.)

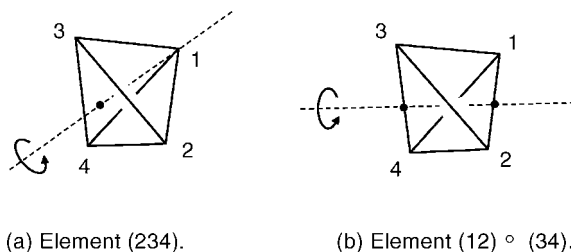


Figure 3.11 Two types of nontrivial elements of rotations of a regular tetrahedron: (a) element (234); (b) element $(12) \circ (34)$ Source: *Modern Algebra with Applications*, 2nd edn., W.J. Gilbert and W.K. Nicholson. © 2004, John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc.

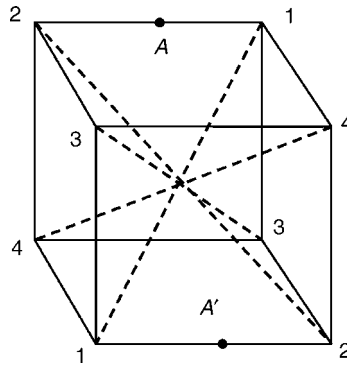


Figure 3.12 The diagonals of a cube that will be axes of rotation

Proof: The regular octahedron is dual to the cube, so it has the same rotation group. There are four diagonals in a cube that join opposite vertices. Label these diagonals 1, 2, 3, and 4 as in Figure 3.12. Any rotation of the cube will permute these diagonals, and this defines a group morphism $f : G \rightarrow S_4$, where G is the rotation group of the cube. \square

The rotation around any vertex of the cube is a cyclic group of order 3 that permutes the three adjacent vertices. The orbit of any vertex is the set of eight vertices. Hence, also by Theorem 3.7, $|G| = 3 \cdot 8 = 24$.

Consider the rotation of order 2 about the line joining A to A' in Figure 3.12. The corresponding permutation is the transposition (12). Similarly, any other transposition is in Image of f . Therefore, "Image of f " = S_4 .

By the morphism theorem discussed in Section 3.3, $G/\text{Kernel } f$ is isomorphic to S_4 and $|G|/|\text{Ker } f| = |S_4|$. Since $|G| = |S_4| = 24$, it follows that $|\text{Ker } f| = 1$, and f is an isomorphism.

The symmetric group S_4 is sometimes called the *octahedral group*.

Theorem 3.14. *The group of proper rotations of a regular dodecahedron and a regular icosahedron is isomorphic to A_5 .*

Proof: A regular dodecahedron is dual to the icosahedron, so it has the same rotation group.

There are 30 edges of an icosahedron, and there are 15 lines through the center joining the midpoints of opposite edges. (The reflection of each edge in the center of the icosahedron is a parallel edge, called the opposite edge.) Given any one of these 15 lines, there are exactly two others that are perpendicular both to the first line and to each other. We call three such mutually perpendicular lines a triad. The 15 lines fall into five sets of triads. Label these triads 1, 2, 3, 4, and 5. Figure 3.13 shows the top half of an icosahedron, where we have labeled the endpoints of each triad. (The existence of mutually perpendicular triads and the labeling of the diagram can best be seen by actually handling a model of the icosahedron.)

A rotation of the icosahedron permutes the five triads among themselves, and this defines a group morphism $f : G \rightarrow S_5$, where G is the rotation group of the icosahedron.

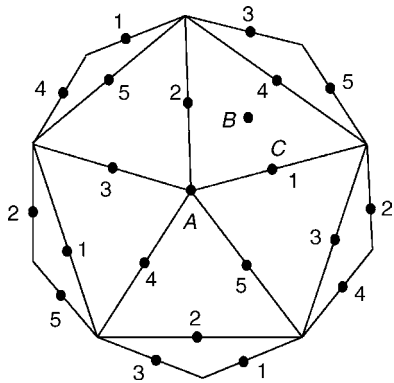


Figure 3.13 The ends of the triads of the icosahedron
Source: Modern Algebra with Applications, 2nd edn., W.J. Gilbert and W.K. Nicholson. © 2004, John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc.

The stabilizer of any vertex of the icosahedron is a group of order 5 that cyclically permutes the five adjacent vertices. The orbit of any vertex is the set of all 12 vertices. Hence, by Theorem 3.11, $|G| = 5 \cdot 12 = 60$.

There are three types of nontrivial elements in G . There are rotations of order 5 about axes through a vertex. The rotations about the vertex A in Figure 3.13 correspond to multiples of the cyclic permutation (12345) , all of which are even. There are rotations of order 3 about axes through the center of a face. The rotations about an axis through the point B , in Figure 3.13, are multiples of (142) and are therefore even permutations. Finally, there are rotations of order 2 about the 15 lines joining midpoints of opposite edges. The permutation corresponding to a rotation about an axis through C , in Figure 3.13, is $(23) \circ (45)$, which is even.

Every 3-cycle occurs in the image of f , so “Image of f ” = A_5 . Since G and A_5 both have 60 elements, the morphism theorem implies that G is isomorphic to A_5 . \square

The alternating group A_5 is sometimes called the *icosahedral group*.
The isomorphic groups to the rotations of a regular polyhedron are summarized in Table 3.2.

Table 3.2 The group of proper rotations of a regular polyhedron and its isomorphism

Polyhedron	Number of members	Isomorphic group
Tetrahedron	12	A_4
Cube	24	S_4
Octahedron	24	S_4
Dodecahedron	60	A_5
Icosahedron	60	A_5

3.7.2 Finite Rotation Groups in Three Dimensions

We will now proceed to show that the only finite proper rotation groups in three dimensions are the three symmetry groups of the regular solids, A_4 , S_4 , and A_5 , together with the cyclic group C_n (given in Example 3.23) and the dihedral group D_n , which is the group of *all* symmetries (both proper and improper rotations) of the regular n -gon.

The unit sphere $S^2 = \{x \in \mathbb{R}^3 \mid \|x\| = 1\}$ is mapped to itself by every element of the orthogonal group $O(3)$. Every rotation group fixing the origin is determined by its action on the unit sphere S^2 . By Theorem 3.11, every nonidentity element $A \in SO(3)$ leaves precisely two antipodal points on S^2 fixed. That is, there exists $x \in S^2$ such that $A(x) = x$ and $A(-x) = -x$. The points x and $-x$ are called the *poles* of A . Let P be the set of poles of the nonidentity elements of a finite subgroup G of $SO(3)$.

Proposition 3.15. *G acts on the set, P , of poles of its nonidentity elements.*

Proof: We show that G permutes the poles among themselves. Let A, B be nonidentity elements of G , and let x be a pole of A . Then $(BAB^{-1})B(x) = BA(x) = B(x)$, so that $B(x)$ is a pole of BAB^{-1} . Therefore, the image of any pole is another pole, and G acts on the set of poles. \square

We classify the rotation groups by considering the number of elements in the stabilizers and orbits of the poles. Recall that the stabilizer of a pole x , $\text{Stab } x = \{A \in G \mid A(x) = x\}$, is a subgroup of G , and that the orbit of x , $\text{Orb } x = \{B(x) \mid B \in G\}$, is a subset of the set P of poles. This is well summarized in Table 3.3, taken from Gilbert and Nicholson [33]. In the table, we can look at the stabilizers and orbits of the poles of the rotation groups that we have already discussed.

We take C_n to be the rotation group of a regular n -agonal cone whose base is a regular n -gon. (The sloping edges of the cone must not be equal to the base edges if $n = 3$.) D_n is the rotation group of a regular n -agonal cylinder whose base is a regular n -gon. (The vertical edges must not be equal to the base edges if $n = 4$.)

Each stabilizer group, $\text{Stab } x$, is a cyclic subgroup of rotations of the solid about the axis through x . The orbit of x , $\text{Orb } x$, is the set of poles of the same type as x . As a check on the number of elements in the stabilizers and orbits, we have $|G| = |\text{Stab } x| |\text{Orb } x|$ for each pole x .

For example, the cube has three types of poles and four types of nontrivial elements in its rotation group; these are illustrated in Figure 3.14.

Now, we have our final conclusion on the rotation groups in three dimensions.












Theorem 3.16. *Any finite subgroup of $SO(3)$ is isomorphic to one of*

















$$C_n \ (n \geq 1), \ D_n \ (n \geq 2), \ A_4, \ S_4 \text{ or } A_5.$$

3.8 Rings

Groups are algebraic systems with one internal composition law. More complicated (and hence, richer) systems are obtained by introducing a second internal composition law that is related to the first. This is what we are going to do in this section.

Table 3.3 The poles of the finite rotation groups (*Source: Modern Algebra with Applications*, 2nd edn., W.J. Gilbert and W.K. Nicholson. © 2004, John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc.)

Group $G =$ $ G =$ Symmetries of	C_n n n -agonal cone 	D_n $2n$ n -agonal cylinder 	A_4 12 tetrahedron 			
Looking down on the pole x	 	  	  			
$ \text{Stab } x =$ $ \text{Orb } x =$	n 1	n 1	2 n	2 6	3 4	3 4

Group $G =$ $ G =$ Symmetries of	S_4 24 cube or octahedron  	A_5 60 dodecahedron or icosahedron  				
Looking down on the pole x	   or   	   or   				
$ \text{Stab } x =$ $ \text{Orb } x =$	2 12	3 8	4 6	2 30	3 20	5 12

3.8.1 Definitions and Examples

The very first algebraic system with two internal composition laws that we are going to consider now is called a *ring*. The abstract concept of a group has its origins in the set of “transformations,” or “permutations” of a set onto itself. In contrast, the idea of a ring stems from another, more familiar, source – the set of integers with two operations, addition and multiplication. Other algebraic systems with two internal composition laws can be obtained by imposing further restrictions on rings.

Definition 3.13: An algebract system $(R, *, \circ)$ is called a *ring* if the internal composition laws $*$ and \circ on the set R satisfy the following three properties:

1. $(R, *)$ is an Abelian group.

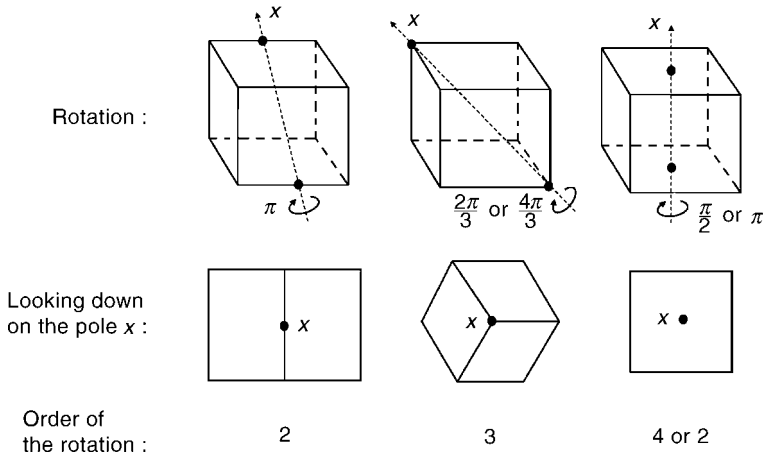


Figure 3.14 The rotations of the cube around the pole x

Source: *Modern Algebra with Applications*, 2nd edn., W.J. Gilbert and W.K. Nicholson. © 2004, John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc.

2. (R, \circ) is a semigroup.
3. The second law \circ is distributive over $*$; that is, for any $a, b, c \in R$,

$$a \circ (b * c) = (a \circ b) * (a \circ c)$$

and

$$(b * c) \circ a = (b \circ a) * (c \circ a).$$

Note: We must assume “both” distributive laws, since the law \circ may not be commutative, in general. □

Example 3.27 (familiar number systems and rings). The familiar examples of rings are the sets of integers, real numbers, rational numbers, even numbers, and complex numbers under the operations of addition and multiplications. This means that $(\mathbb{Z}, +, \cdot)$, $(\mathbb{Q}, +, \cdot)$, $(\mathbb{R}, +, \cdot)$, $(E, +, \cdot)$, $(\mathbb{C}, +, \cdot)$ are all rings.

It is also possible to devise rings by taking help of the familiar rings. Let us consider the set $\mathbb{Z}[\sqrt{3}]$ of all numbers of the form $a + b\sqrt{3}$, where $a, b \in \mathbb{Z}$ and $a + b\sqrt{3} = c + d\sqrt{3}$ if and only if $a = c$ and $b = d$. Let us define addition \oplus and multiplication \odot of two such numbers in terms of ordinary addition and multiplication as follows:

$$(a + b\sqrt{3}) \oplus (c + d\sqrt{3}) = (a + c) + (b + d)\sqrt{3},$$

$$(a + b\sqrt{3}) \odot (c + d\sqrt{3}) = (a \cdot c + 3 \cdot b \cdot d) + (a \cdot d + b \cdot c)\sqrt{3}.$$

Check that $(\mathbb{Z}[\sqrt{3}], \oplus, \odot)$ is a ring. □

Because of all these familiar examples of rings, it is customary to call the first composition law “addition” and denote it by $+$, and to call the second law “multiplication” and denote it by \cdot . Thus we express a ring as $(R, +, \cdot)$ instead of $(R, *, \circ)$, although $+$ and \cdot may not necessarily mean addition and multiplication. In keeping with the convention, we shall also

refer to the identity of $(R, +)$ as the *additive identity* and denote it by 0 (zero). Finally, the “additive inverse” of an element $a \in R$ will be denoted by $(-a)$. Because of the uniqueness of 0 and $(-a)$, we usually write $a - b$ instead of $a + (-b)$. We again emphasize at this point that the use of these terminologies does not mean that the composition laws $+$ and \cdot on the ring $(R, +, \cdot)$ have all the properties that $+$ and \cdot have in the system of real numbers.

In what follows, we will present a few more examples of rings.

Example 3.28 (rings in shapes). In Example 3.19 we have shown that the algebraic system $(\mathcal{P}(\mathbf{R}^3), \Delta)$ (where Δ denotes symmetric difference between two sets) is an Abelian group. We also know that the algebraic system $(\mathcal{P}(\mathbf{R}^3), \cap)$ is a monoid (see Example 3.14). Now let $A, B, C \in \mathcal{P}(\mathbf{R}^3)$. It can be shown that

$$A \cap (B \Delta C) = (A \cap B) \Delta (A \cap C),$$

and

$$(B \Delta C) \cap A = (B \cap A) \Delta (C \cap A).$$

Therefore, $(\mathcal{P}(\mathbf{R}^3), \Delta, \cap)$ is a ring. The additive identity here is the null set \emptyset and the additive inverse of A is A itself. \square

Example 3.29 (congruence class). In Section 2.4.2, we discussed the arithmetical notion of congruence. Let \mathbb{Z} be the set of integers and let \mathbb{Z}_m be the set of equivalence classes generated by the equivalence relation “congruence modulo in” for any positive integer m . We say that two integers a and a' are congruent modulo in m – that is, $a' \equiv a(\text{mod } m)$ – if and only if $(a' - a)/m$ is an integer. The collection of all integers that are congruent to a given integer a is generally denoted by $[a]$ (see Section 2.4.2). Thus,

$$[a] = \{ a' \mid a' \equiv a(\text{mod } m), a \in \mathbb{Z} \}.$$

The algebraic systems $(\mathbb{Z}_m, +_m)$ and (\mathbb{Z}_m, \times_m) are the Abelian group and the monoid, respectively, in which the operations $+_m$ and \times_m are defined in terms of the operations $+$ (addition) and \cdot (multiplication) on \mathbb{Z} as follows. For any $[a]$ and $[b]$,

$$[a] +_m [b] = [a + b],$$

$$[a] \times_m [b] = [a \cdot b].$$

The algebraic system $(\mathbb{Z}_m, +_m, \times_m)$ can be shown to be a ring. The additive identity here is only $[0]$.

For further details, see Mac Lane [63]. \square

One fundamental question may arise at this point. “Since the ring is a direct generalization of our concept of the integers, are the familiar facts from elementary algebra (in other words, the familiar arithmetic facts) valid in a ring?” Using the definition of the ring $(\mathbb{Z}, +, \cdot)$, in which 0 is the additive identity and $-a$ denotes the additive inverse of an element $a \in \mathbb{Z}$,

- (i) $a \cdot 0 = 0 \cdot a = 0$,
- (ii) $a \cdot (-b) = (-a) \cdot b = -(a \cdot b)$,
- (iii) $(-a) \cdot (-b) = a \cdot b$,
- (iv) $(-1) \cdot (-a) = -a$,
- (v) $(-1) \cdot (-1) = -1$.

These facts can be easily proved from the ring definition as a whole.

The above results conform to the familiar arithmetic facts. However, this does not mean that all the arithmetic facts to which we have become accustomed hold for general rings. For example, from (i), $a \cdot 0 = 0$ for any a ; that is, the product of two elements in a ring is zero whenever either factor is zero. Surprisingly enough, its converse is false. It can happen (and it often does happen) that the product of two nonzero elements in a ring is zero. The simplest examples of this phenomenon are found in rings \mathbb{Z}_m when a and m are not prime numbers. For instance, assume that $m = r \cdot s$, where r and s are integers such that $1 < r < m$ and $1 < s < m$. Now, $[r] \neq [0]$ and $[s] \neq [0]$, but $[r] \times_m [s] = [r \cdot s] = [m] = [0]$. This means that the product of two nonzero elements happens to be zero in our familiar arithmetic.

Other differences can also be found. Generally speaking, it is possible to add, subtract, and multiply elements in a ring, but it is not always possible to divide. Even in an integer domain, where elements can be canceled, it is not always possible to divide by nonzero elements. For example, if $x, y \in \mathbb{Z}$, then $2x = 2y$ implies that $x = y$, but not all elements in \mathbb{Z} can be divided by 2. This means that it is not possible to divide within a general ring.

You may have realized at this point that the nature of a ring $(R, +, \cdot)$ is essentially characterized by its structure. Thus, depending upon the structure, various special cases of rings are defined as follows.

3.8.2 Some Classes of Rings

3.8.2.1 Commutative Rings

If (R, \cdot) is commutative – that is, $a \cdot b = b \cdot a$ for all $a, b \in R$ – then the ring $(R, +, \cdot)$ is called a “commutative ring.” All of the examples of rings that we have given so far are commutative rings.

Example 3.30. The set of all $n \times n$ square matrices with real coefficients forms a ring $(M_n(\mathbb{R}), +, \cdot)$, which is *not commutative* if $n > 1$, because matrix multiplication is not commutative. \square

Example 3.31. $(\mathbb{Z}_m, +_m, \times_m)$ is a commutative ring, where addition and multiplication on congruence class, modulo m , are defined by the equations $[x] +_m [y] = [x + y]$ and $[x] \times_m [y] = [x \cdot y]$. \square

3.8.2.2 Rings with Identity

If (R, \cdot) is a monoid – that is, it has an identity element for multiplication – then $(R, +, \cdot)$ is called a “ring with identity.” We denote this multiplicative identity by 1, so that $1 \cdot a = a \cdot 1 = a$ for all $a \in R$. We recall that if 1 exists, it is unique. Note that a ring with identity need not

be commutative and vice versa. For example, the ring $(E, +, \cdot)$ of even integers under usual operations of addition and multiplication is a commutative ring but not a ring with identity.

3.8.2.3 Rings without Divisors of Zero

When we have $a \cdot b = 0$, we then say that both a and b are “divisions of zero.” Now, it is possible to define a “ring without divisors of zero.” In such a ring, $(R, +, \cdot)$ will be closed with respect to the operation \cdot . By “close” we mean, for any a where $a \in R$, that the product $a \cdot 0 = 0 \cdot a = 0$. Conversely, in a ring without divisors of zero, $a \cdot b = 0$ implies $a = 0$ or $b = 0$. The importance of a ring without divisors of zero is that the familiar “cancellation law” of ordinary arithmetic holds.

One very useful property of the familiar number systems is the fact that if $a \cdot b = 0$, then either $a = 0$ or $b = 0$. This property allows us to cancel nonzero elements, because if $a \cdot b = a \cdot c$ and $a \neq 0$, then $a \cdot (b - c) = 0$, so $b = c$. However, this property does not hold for all rings. For example, in \mathbb{Z}_4 , we have $[2] \times_4 [2] = [0]$, and we cannot always cancel, since $[2] \times_4 [1] = [2] \times_4 [3]$, but $[1] \neq [3]$.

3.8.2.4 The Integral Domain

If $(R, +, \cdot)$ is a commutative ring, a nonzero element $a \in R$ is called a *zero divisor* if there exists a nonzero element $b \in R$ such that $a \cdot b = 0$. A nontrivial commutative ring is called an “integral domain” if it has no zero divisors. Hence a nontrivial commutative ring is an integral domain if $a \cdot b = 0$ always implies that $a = 0$ or $b = 0$.

As the term implies, the integers form an integral domain. Also, \mathbb{Q} , \mathbb{R} , and \mathbb{C} are integral domains. However, \mathbb{Z}_4 is not, because $[2]$ is a zero divisor. Nor is $(\mathcal{P}(X), \Delta, \cap)$, because every nonempty proper subset of X is a zero divisor.

Example 3.32. $M_n(\mathbb{R})$ is not an integral domain: for example,

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^2 = O.$$

□

3.8.2.5 Division Rings

We stated that in the case of a ring without divisors of zero, the system $(R, +, \cdot)$ is closed with respect to the operation. But this does not mean that (R, \cdot) will be a group. For example, consider the familiar ring of integers $(\mathbb{Z}, +, \cdot)$. This is a ring without divisors of zero and it has the identity (multiplicate), which is the number 1. However, the system (\mathbb{Z}, \cdot) does not form a group, since except for the number 1, no other elements of the \mathbb{Z} have a multiplicative inverse within the set. Thus we can define a new type of ring where the system (R, \cdot) is a group. Such a ring is called a “division ring.” In other words, a ring is said to be a group under multiplication. (Some mathematicians use the terminology “skew field” instead of “division ring.”) Therefore, it follows from our previous discussion that \mathbb{Z} is not a division ring. However,

our other familiar algebraic systems of rational numbers \mathbb{Q} , real numbers \mathbb{R} , and complex numbers \mathbb{C} with ordinary addition and multiplication are all division rings. (At this point, we can justify the use of the term “division ring” in the following way: within these systems, we can consider the inverse a^{-1} of an element a as $a \cdot a^{-1} = a^{-1} \cdot a = 1$, where 1 denotes multiplicative identity.) Note the following interesting fact. Although the ring $(\mathbb{Z}_m, +_m, \times_m)$ is not necessarily a division ring, it will be a division ring if m is a prime integer. For each of the nonzero elements of division ring, the inverse exists. For example, in \mathbb{Z}_7 , the inverse of the element [5] is [3], in \mathbb{Z}_7 , the inverse of [4] is [3], and so on.

3.8.2.6 The Field

Finally, we can come to the definition of a very important algebraic system known as a *field*.

Definition 3.14: A “field” is a commutative division ring. □

The division rings that we have mentioned earlier – that is, the rings \mathbb{R} or \mathbb{C} – are all commutative division rings. Therefore, they all are fields. Roughly speaking, fields are the “number systems” of mathematics. However, there are division rings that are noncommutative. We shall give below an example of a noncommutative ring known as a “quaternion.” The extreme importance of the quaternion has been highly evident in the discipline of shape description, in modeling of the rotation of objects in space.

3.8.3 The Ring of Quaternions and Rotation of Objects

The example that we will present now is often called the *ring of quaternions*. This ring was first discovered by the Irish mathematician Sir William Rowan Hamilton, on Monday, October 16, 1843 (one of the best documented days in the history of mathematics). Initially, it was extensively used in the study of mechanics, but for some mysterious reasons it disappeared in history for about 100 years. In recent times, it has again been used extensively in the world of aerospace engineering, robotics, and computer graphics. In the field of application of quantum physics, quaternions are known as Clifford algebras [3], after the English mathematician William Kingdom Clifford, who recognized their importance.

A quaternion \mathbf{q} is defined in the following way:

$$\mathbf{q} \equiv q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}. \quad (3.4)$$

where q_0, q_1, q_2, q_3 are real numbers and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are three imaginary units.

Let \mathcal{Q} denote the set of all the quaternions. In order to make \mathcal{Q} into a ring, we must define an appropriate $+$ and \cdot for its elements. To this end, for any $\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ and $\mathbf{q}' = q'_0 + q'_1\mathbf{i} + q'_2\mathbf{j} + q'_3\mathbf{k}$ in \mathcal{Q} :

1. The addition $+$ of two quaternions is as follows:

$$\mathbf{q} + \mathbf{q}' = (q_0 + q'_0) + (q_1 + q'_1)\mathbf{i} + (q_2 + q'_2)\mathbf{j} + (q_3 + q'_3)\mathbf{k}.$$

2. The product \cdot of any two of the quaternions is defined by the requirement that $1 = 1 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$ acts as the multiplicative identity and by the table

$$\begin{cases} \mathbf{i}^2 = \mathbf{j}^2 = -1, \\ \mathbf{i} \cdot \mathbf{j} = -\mathbf{j} \cdot \mathbf{i} = \mathbf{k}. \end{cases}$$

Thus

$$\begin{aligned} \mathbf{q} \cdot \mathbf{q}' &= (q_0q'_0 - q_1q'_1 - q_2q'_2 - q_3q'_3) \\ &\quad + (q_0q'_1 + q_1q'_0 - q_2q'_3 + q_3q'_2) \mathbf{i} \\ &\quad + (q_0q'_2 + q_1q'_3 + q_2q'_0 - q_3q'_1) \mathbf{j} \\ &\quad + (q_0q'_3 - q_1q'_2 + q_2q'_1 + q_3q'_0) \mathbf{k}. \end{aligned}$$

Note that, from the definition of \cdot ,

$$\begin{aligned} \mathbf{k}^2 &= (\mathbf{i} \cdot \mathbf{j}) \cdot (\mathbf{i} \cdot \mathbf{j}) = -\mathbf{i}^2 \cdot \mathbf{j}^2 = -1, \\ \mathbf{k} \cdot \mathbf{i} &= (\mathbf{i} \cdot \mathbf{j}) \cdot \mathbf{i} = -\mathbf{i}^2 \cdot \mathbf{j} = \mathbf{j}. \end{aligned}$$

(For simplicity, the multiplication of quaternions, $\mathbf{a} \cdot \mathbf{b}$, can be written as \mathbf{ab} , as long as this will not cause any confusion.)

It is easy to prove that $(\mathcal{Q}, +)$ is an Abelian group in which the additive identity 0 is $0 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$.

Quaternion multiplication is seen to be noncommutative, because $\mathbf{qq}' \neq \mathbf{q}'\mathbf{q}$. The element 1 serves as the multiplicative identity. The proof that the nonzero quaternions form a group under quaternion multiplication is simple, except for the existence of multiplicative inverses. To that end, we define the “conjugate” of a quaternion \mathbf{q} as

$$\bar{\mathbf{q}} = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}.$$

If $\mathbf{q} = q_0 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$, then $\bar{\mathbf{q}} = \mathbf{q}$.

Hereafter, we also denote $\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ as $\mathbf{q} = (q_0, q_1, q_2, q_3)$.

$\mathbf{q}\bar{\mathbf{q}} = (q_0^2 + q_1^2 + q_2^2 + q_3^2, 0, 0, 0)$ becomes scalar. So, let us define the “norm” of \mathbf{q} as $|\mathbf{q}| \equiv (\mathbf{q}\bar{\mathbf{q}})^{\frac{1}{2}} = (q_0^2 + q_1^2 + q_2^2 + q_3^2)^{\frac{1}{2}}$. In this notation, \mathbf{q} has the inverse $\mathbf{q}^{-1} = \bar{\mathbf{q}}/(\mathbf{q}\bar{\mathbf{q}}) = \bar{\mathbf{q}}/|\mathbf{q}|^2$. Therefore, (\mathcal{Q}, \cdot) is a noncommutative group, and hence $(\mathcal{Q}, +, \cdot)$ turns out to be a noncommutative division ring.

For our present purpose, we are particularly interested in the geometric interpretation of a quaternion. Just as rectangular coordinates represent the position of an object as a single vector, a quaternion represents the orientation of an object as a single entity. Just as a number of translations of an object can be combined by adding corresponding vectors, various rotations of an object can be combined by multiplying the corresponding quaternions. Below, we briefly present the relationship between quaternions and rotations, without going into the mathematical details (for further study, see Altmann [1]).

Let us consider the rotation of a point in three-dimensional space by an angle θ around some arbitrary axis. We depict such a situation in Figure 3.15, where the point is represented by the vector \mathbf{p} and the axis of rotation is specified by the unit vector \mathbf{n} . Assume that the point \mathbf{p} has moved to \mathbf{p}' due to this rotation. The following equation gives the transformed

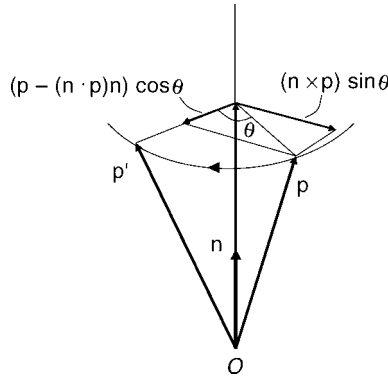


Figure 3.15 The rotation of a point represented by the vector \mathbf{p} by an angle θ around the axis \mathbf{n}

point \mathbf{p}' , in terms of the initial point \mathbf{p} , the angle of rotation θ , and the direction of the axis of rotation \mathbf{n} :

$$\mathbf{p}' = (\mathbf{n} \cdot \mathbf{p})\mathbf{n} + (\mathbf{p} - (\mathbf{n} \cdot \mathbf{p})\mathbf{n}) \cos \theta + (\mathbf{n} \times \mathbf{p}) \sin \theta, \quad (3.5)$$

where $+$, \times , and \cdot denote vector addition, the vector cross-product, and the vector dot product, respectively. (We assume here that you are familiar with basic vector algebra. The derivation of the above equation can be obtained in any standard textbook on computer graphics; for example, Rogers and Adams [86].)

Let us try to find the transformed point \mathbf{p}' with the help of quaternion algebra.

To achieve our task, we first have to express a vector \mathbf{p} in the quaternion language. Let us express \mathbf{p} as the following quaternion:

$$\mathbf{P} = 0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k},$$

where x, y, z are the coordinates of the point and components of the vector \mathbf{p} . In compact notation, \mathbf{P} can be written as $(0, x, y, z)$.

Now consider a quaternion $\mathbf{P}' = (0, x', y', z')$ of another point \mathbf{p} . Then,

$$\begin{aligned} \mathbf{P}\mathbf{P}' &= (-xx' - yy' - zz', \\ &\quad yz' - zy', zx' - xz', xy' - yx') \end{aligned} \quad (3.6)$$

The first term of this quaternion product is the negative of the inner product of two positional vectors \mathbf{p} and \mathbf{p}' , and the remaining terms are the components of their outer product. When the points \mathbf{p} and \mathbf{p}' are on a unit sphere E , as shown in Figure 3.16, and the angle $\angle \mathbf{pO}\mathbf{p}'$, which is an angle along with a great circle passing through \mathbf{p} and \mathbf{p}' , is denoted as θ , the inner and outer products of \mathbf{p} and \mathbf{p}' are

$$\mathbf{p} \cdot \mathbf{p}' = \cos \theta, \quad (3.7)$$

$$\mathbf{p} \times \mathbf{p}' = \overrightarrow{OQ} \sin \theta, \quad (3.8)$$

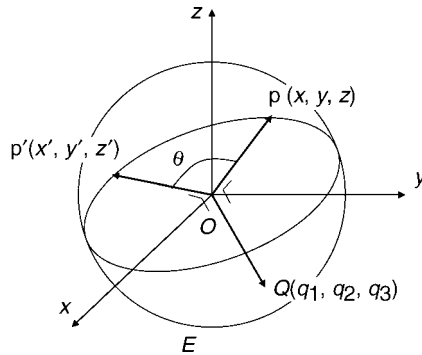


Figure 3.16 The rotation of a point on a unit sphere

respectively, where Q is the point on the unit sphere and \overrightarrow{OQ} is perpendicular to the great circle containing \mathbf{p} and \mathbf{p}' . Denoting the three-dimensional coordinates of Q by (q_1, q_2, q_3) , from $|\overrightarrow{OQ}| = 1$, the product of (3.6) is given as

$$\mathbf{P}\mathbf{P}' = (-\cos \theta, q_1 \sin \theta, q_2 \sin \theta, q_3 \sin \theta). \quad (3.9)$$

Note that, when $\theta = 0$, then $\mathbf{P}' = \mathbf{P}$, and

$$\mathbf{P}^2 = -1.$$

According to Rodrigues (see [1]), the quaternion corresponding to the rotation θ around the axis \overrightarrow{OQ} is

$$\mathbf{q} = \left(\cos \left(\frac{\theta}{2} \right), q_1 \sin \left(\frac{\theta}{2} \right), q_2 \sin \left(\frac{\theta}{2} \right), q_3 \sin \left(\frac{\theta}{2} \right) \right), \quad (3.10)$$

where q_1, q_2, q_3 are the components of a unit vector \overrightarrow{OQ} .

Note that the norm of the quaternion $|\mathbf{q}| = 1$ in this case. A quaternion of unit norm is called a “normalized quaternion.”

Proposition 3.17 (rotation in the quaternion representation). *The rotation \mathbf{p}' of \mathbf{p} around an axis \overrightarrow{OQ} through an $\angle \theta$ is given by*

$$\mathbf{P}' = \mathbf{q}\mathbf{P}\bar{\mathbf{q}}, \quad (3.11)$$

where \mathbf{q} is a quaternion of (3.10). Equation (3.11) is valid when \mathbf{p} and \mathbf{p}' are not on a unit circle.

You are invited to check the proposition by applying the quaternion multiplication rule and comparing the result with (3.5).

In summary, when vector \mathbf{p} is rotated into vector \mathbf{p}' , the quaternion \mathbf{P} (corresponding to \mathbf{p}) gets changed into the quaternion \mathbf{P}' , which is the “conjugate of \mathbf{P} under \mathbf{q} (the normalized quaternion corresponding to the required rotation,” given as (3.11).

Conversely, when we are given \mathbf{p} and \mathbf{p}' , the axis and the angle of rotation are derived by means of the following steps.

Let \mathbf{P} and \mathbf{P}' denote corresponding quaternions of the *normalized* \mathbf{p} and \mathbf{p}' , respectively. From (3.9),

$$\mathbf{q}^2 = -\overline{\mathbf{P}\mathbf{P}'} = -\overline{\mathbf{P}'}\overline{\mathbf{P}} = -\mathbf{P}'\mathbf{P}.$$

Therefore, being given $-\mathbf{P}'\mathbf{P} = (c_0, c_1, c_2, c_3)$ and $\mathbf{q} = (r_0, r_1, r_2, r_3)$, we have

$$\begin{aligned} r_0 &= \pm \sqrt{(1 + c_0)/2}, & r_1 &= c_1/(2r_0), \\ r_2 &= c_2/(2r_0), & r_3 &= c_3/(2r_0). \end{aligned}$$

In this form, \pm means that $+$ gives the rotation along the shorter arc of the great circle periphery and $-$ gives the rotation along the longer arc. It should be noted that \mathbf{p} and \mathbf{p}' are given as symmetrical points with respect to the origin: we have no proper rotation for these points.

The advantage of the quaternion over the vector can be seen when composite rotation has to be determined. Complicated mechanisms can often be analyzed by breaking them down into elementary sections, such as the two-bar robot arm shown in Figure 3.17. Here, O_1 is fixed, O_1QO_2 is a rigid arm such that $\angle O_1QO_2 = 90^\circ$, and $|O_1Q| = |QO_2| = 1$ (say). If we define axes (x, y, z) as shown at O_1 , then the arm is free to rotate (angle θ_1) at O_1 about the z -axis. Also, the rod QO_2P can rotate (angle θ_2) at O_2 about the y -axis. Suppose that we want an express position of the end-effector P relative to O_1 .

In this case, the rotation of θ_1 about z is described by $\mathbf{q}_1 = (\cos(\theta_1/2), 0, 0, \sin(\theta_1/2))$, and the rotation of θ_2 about y is $\mathbf{q}_2 = (\cos(\theta_2/2), 0, \sin(\theta_2/2), 0)$ (we can define any other rotation axis). The quaternion \mathbf{X}_1 corresponds to $\overrightarrow{O_1O_2} = \mathbf{x}_1$ initially, and \mathbf{X}_2 to $\mathbf{x}_2 = \overrightarrow{O_2P}$. Then the quaternion representing the position of P relative to O_1 is given as $\mathbf{X}_p = \mathbf{X}_1 + \mathbf{X}_2$.

First, we consider that, by the second rotation of θ_2 , the respective positional quaternions change to

$$\mathbf{X}'_2 = \mathbf{q}_2\mathbf{X}_2\overline{\mathbf{q}_2},$$

$$\mathbf{X}'_p = \mathbf{X}_1 + \mathbf{X}'_2.$$

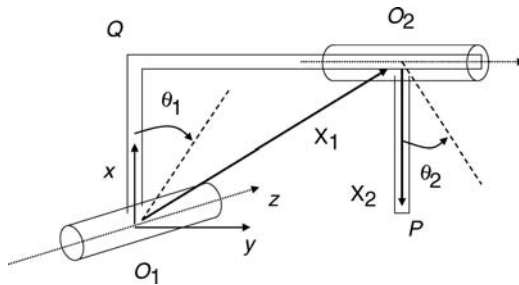


Figure 3.17 A simple two-bar robot arm with two degrees of freedom

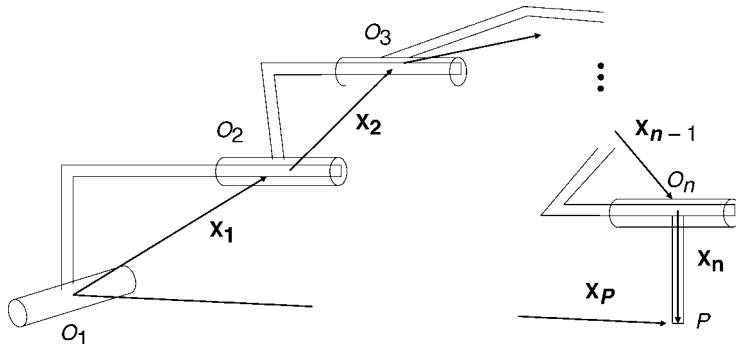


Figure 3.18 A sketch of an n -bar robot arm

Now, consider the first rotation of θ_1 . \mathbf{X}'_p changes further, to

$$\mathbf{X}''_p = \mathbf{q}_1 \mathbf{X}'_p \overline{\mathbf{q}_1}.$$

As the result, the end-effector point finally changes to

$$\mathbf{X}''_p = \mathbf{q}_1 \mathbf{X}_1 \overline{\mathbf{q}_1} + \mathbf{q}_1 \mathbf{q}_2 \mathbf{X}_2 \overline{\mathbf{q}_2} \overline{\mathbf{q}_1}.$$

This form can easily be extended to an n -bar robot arm, an example of which is shown in Figure 3.18.

For this case, the position of the end-effector can be written down immediately as

$$\mathbf{X}_p = \mathbf{q}_1 \mathbf{X}_1 \overline{\mathbf{q}_1} + \mathbf{q}'_2 \mathbf{X}_2 \overline{\mathbf{q}'_2} + \mathbf{q}'_3 \mathbf{X}_3 \overline{\mathbf{q}'_3} + \cdots + \mathbf{q}'_n \mathbf{X}_n \overline{\mathbf{q}'_n}, \quad (3.12)$$

where $\mathbf{q}'_i = (\mathbf{q}_1 \cdots \mathbf{q}_{i-1})$, for $i = 2, 3, \dots, n$, and \mathbf{X}_i denotes the initial positional quaternion of O_{i+1} relative to O_i .

4

Morphological Models for Shape Description and Minkowski Operators

4.1 The Objective of Shape Description Modeling

In this chapter, a simple shape model is presented, using two basic shape operators called the Minkowski addition and decomposition operators. The mathematical characteristics of these operators are explored in some detail, with the aim of eventually arriving at a formal theory of shape description. A few areas of application of the shape model – especially, some important uses of the shape operators – are briefly mentioned. Note that image analysis using these shape operators is called *mathematical morphology*, and we will also sum up the operators from this point of view.

The significance of the Minkowski operators (vector addition and its inverse) can be compared with that of the set operators – especially when we are dealing with the shapes of objects and various interactions among objects in space. We propose a shape description model using the Minkowski addition (vector addition) and Minkowski decomposition (the inverse of vector addition) operators as “shape operators.” The mathematical characteristics of these shape operators are investigated in order to build up a formal theory of the proposed model. To demonstrate the usefulness of the model, and particularly the significance of the shape operators, we will briefly mention a few areas of application.

The motivation for this work stems from the fact that there is a distinct lack of, and an urgent need for, a rigorously developed formal theory of shape description that would deal with the underlying principles of shape representation, shape analysis, and various operations on the shapes of objects. (Note that the term “description” has been used here in a broader sense than just “representation.”) There are a number of important issues regarding shape description that we face frequently, the first set being concerned with the shape of a single object. Let us list a few examples:

- How can we formally decide whether a given shape representation scheme can supply “complete” shape information?
- Even if a representation scheme is complete, how can we determine “what aspects” of the information are important for a given set of applications, and whether or not “those aspects” are explicit in that particular representation scheme? It is important to note that, even if a representation scheme is complete, it makes certain information explicit at the expense of information that is pushed into the background, and may be quite difficult – though not impossible – to extract.
- What class of shapes is the representation/analysis scheme designed for, and do the shapes in that class have canonical descriptions in the scheme?

The other set of issues is related to the question of interaction between two or more objects. For example:

- Interactions among objects, such as intersections between two objects, the containment of one object inside another, the distance between two objects, and so on, are frequently encountered in practice. How can we formally evaluate a description scheme in terms of its performance in carrying out these interactions?
- Is it possible to classify shapes according to their computational complexities in carrying out such interaction tasks?

This list of issues can be extended further, but because of the lack of formal theory, in most of these situations there are no clear-cut answers to these questions; approaches to shape description are usually highly intuitive and *ad hoc* responses to practical needs.

However, in this book we do not claim to present a formal theory of the proposed shape model in a complete form. Our attempt could be considered as an approach toward formalization of a shape description model. This chapter attempts to bring out the importance of studying the nature of shape operators in such formalizations and, in particular, it demonstrates the usefulness of the Minkowski operators as shape operators.

To achieve our aim, we have organized this chapter into two parts: (a) In first part, we introduce a mathematical model for shape description (Section 4.2). The Minkowski operators are used as the shape operators in the model. The relevant mathematical nature of the Minkowski operators is then studied (Section 4.3). The suitability of the Minkowski operators as shape operators is emphasized and also discussed in some detail in this first part (Section 4.4). (b) The second part of the chapter deals with the practical applications of the shape model and the Minkowski operators. The immediate areas of application are geometric modeling and computer-aided design and graphic arts (Section 4.5), which are concerned with representation of the shapes of three-dimensional and two-dimensional objects. Another important application is in mathematical morphology, which is considered to be an important tool in image processing (Section 4.6). Apart from its applications in the shape representation and analysis problems, a number of problems where the motion of objects among other objects is concerned can be transformed into Minkowski operations, and hence tackled more efficiently than by the existing methods.

4.2 The Basic Idea of Model Description

4.2.1 The Model

We can start with a simple assumption that a model should be “as close as possible” to the mental schema of a human user. The interaction of a user with a model remains substantial, even after a large number of operations on the model have been automated and are carried out by a machine. In order to minimize the computational effort during interaction, it must be easy for a user to develop a conceptual understanding of the model.

Unfortunately, psychologists are as yet unable to tell us “unambiguously” how we visually perceive shapes. One popular approach is a “factor-analytic approach.” A traditional factor-analytic approach would dissect a shape and measure its components, on the assumption that synthesizing the measure would reconstitute the shape. This approach could be formalized in the way discussed in Section 1.5.

In the following, we apply our descriptions to both continuous and discrete shapes. Let us use \mathbb{G} instead of \mathbb{R} and \mathbb{Z} , and decide to work on $E = \mathbb{G}^d$; that is, both d -dimensional real Euclidean space and d -dimensional discrete space. In what follows, for any binary shape, A is a subset of E , given as

$$A = \{a \in A \subset \mathbb{G}^2\}. \quad (4.1)$$

The formal model will be a context-free grammar G , consisting of a 4-tuple

$$G = (T, N, P, S),$$

where T denotes terminal symbols, which contain two sets – a set of “primitive shapes” called K and a set of “shape operators” called $*$; N denotes nonterminal symbols – that is, “complex shapes”; S denotes the start symbol, which is “shape”; and P denotes a finite set of production rules or shape rules in the form

$$\begin{aligned} S &\rightarrow K; & S &\rightarrow N; \\ N &\rightarrow N * N; & N &\rightarrow N * K; & N &\rightarrow K * K. \end{aligned}$$

(The shape rule essentially says that any shape is regarded as a complex shape that can be described in terms of primitive shapes and shape operators.)

$L(G)$ denotes the “language” generated by G . (It must be noted that many of the existing shape representation and shape analysis schemes are modeled as grammars of the above type [21] – although sometimes without stating this explicitly.)

Some of the issues that we raised earlier can now be stated more formally:

1. Each element (say, “sentence”) in $L(G)$ represents a “syntactic” description of some shape. Therefore, the “domain” D of the model – that is, the set of shapes that can be described in this scheme – is defined by $L(G)$. (The usefulness of this obvious formalization will become more apparent when we show later, in Section 4.4, how the domain of our model can be estimated by studying the nature of T .)

Whether or not every element of $L(G)$ will correspond to the shape of some physically realizable object depends entirely on the set T . Note that even if K corresponds to a set of

physically realizable primitive shapes and $*$ is precisely defined on them, a syntactically correct description does not guarantee a physically realizable shape. The nature of $*$ plays a vital role in deciding the physical validity of shapes. There are two ways of handling such a situation: (a) to use shape operators that inherently take care of the geometry and topology of shape, hence guaranteeing the physical validity of the product-shape; or (b) to carry out some kind of regularization on the product-shape to make it valid, and to include this regularization process within the definitions of the shape operators. The “regularized set operators” in the constructive solid geometry (CSG) scheme fall into the second category.

2. More precise meanings can be given to the terms “shape representation” and “shape analysis.” The shape representation task can be viewed as the task of “generating” a sentence in $L(G)$, while shape analysis is the task of “parsing” a sentence.
3. The “equivalence” of two shape description schemes, G and G' , can be formally established by showing that $L(G) = L(G')$.
4. A description scheme is “ambiguous” if $L(G)$ contains an ambiguous sentence. In turn, a sentence is “syntactically ambiguous” if it has more than one canonical derivation; that is, if the derivation is not “unique.” For example, the “sweep representation” scheme can be regarded as syntactically ambiguous, since a rectangular parallelepiped can be represented in several different ways. A sentence is “semantically ambiguous” if, for a given canonical derivation, it has more than one interpretation. The wire-frame representation scheme, for example, can be said to be semantically ambiguous.

4.2.2 The Shape Operator

It appears that the selection of shape operators $*$ is the most important task in defining a shape grammar. At present, there are no formal guidelines for selecting shape operators for a given class of shapes to be described. One way is to carefully choose a set of shape operators and study their suitability for a given set of applications.

For our model, we choose the following shape operators:

$$* = \{ \oplus, \ominus, \odot \},$$

where \oplus denotes the Minkowski addition operator; \ominus denotes the Minkowski decomposition operator, which is the “inverse” of the \oplus operator; and \odot denotes the glue operator, which is a restricted form of union that applies only to shapes with disjoint interiors.

The operators \oplus and \ominus are just the same as those known as the *dilation* and *erosion* operators, respectively, in the field of “morphology.” We will consider morphology briefly at the end of this chapter. The \odot operator has been used in a few other shape representation models. In this chapter, we will concentrate more on the two Minkowski operators in the context of shape description.

Note that $*$ may not be the minimal set. It appears that the subset $\{ \oplus, \odot \}$ has the same power of describing convex shapes as the set $*$. However, in this chapter we do not attempt to show that one of the operators can be derived from the other two.

4.2.2.1 The Minkowski Addition Operator

Minkowski addition can be regarded as a kind of “dilation” or “growing.” A precise definition of Minkowski addition can be given as follows.

Definition 4.1: Let B and T be two arbitrary sets in \mathbb{G}^d space. The resultant set S is obtained by positioning B at every point of T ; that is, by vectorially adding all the points of B to those of T . We denote this by

$$S = B \oplus T = \left\{ c \in \mathbb{G}^d : c = b + t, b \in B, t \in T \right\}, \quad (4.2)$$

where “ \oplus ” stands for *Minkowski addition*. □

Some simple examples in two dimensions, shown in Figure 4.1, may help to clarify the idea. Conceptually, Minkowski addition can be regarded as the growing of an object by another object – as if growing a trajectory T by means of brush B . The first operand B is usually also called the *structuring element*, and is considered to be a parameter of dilation. Figures 4.2 and 4.3 show Minkowski additions as the growing of trajectories by means of brushes. This same analogy can be easily extended to three dimensions, as shown in this figure, and it goes beyond the brush-trajectory metaphor. (Note that in our model, the sweep representation scheme becomes a special case.)

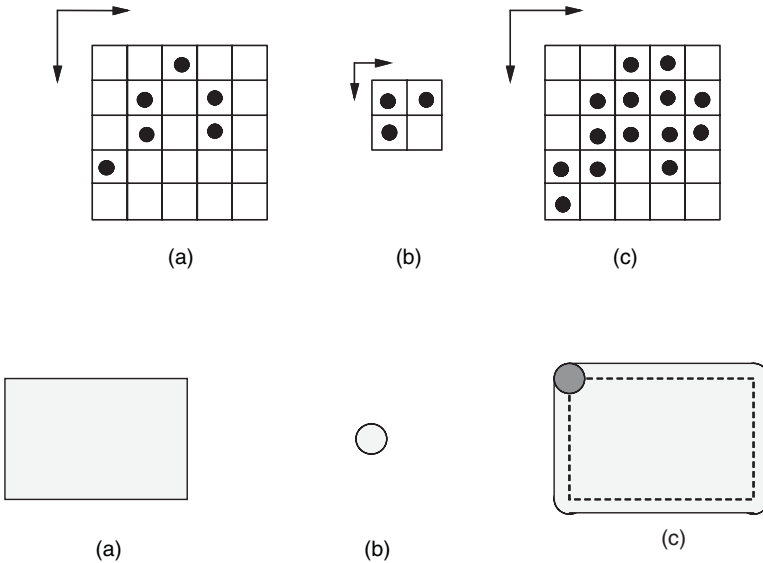


Figure 4.1 Minkowski addition in the discrete domain (top row) and in the continuous domain (bottom row): (a) the input shape T ; (b) the structuring element B ; (c) their Minkowski addition of $S = B \oplus T$

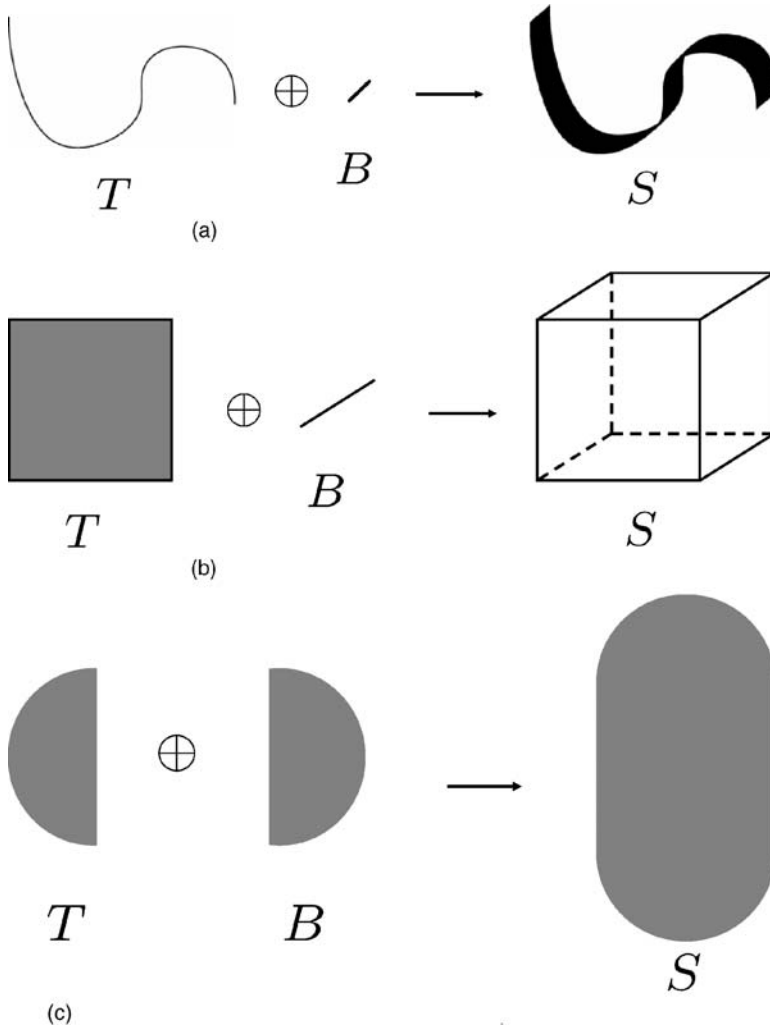


Figure 4.2 Minkowski addition as the growing of a trajectory T by a brush B : (a) Minkowski addition in two dimensions; (b) Minkowski addition in three dimensions; (c) Addition goes beyond the brush-trajectory metaphor

4.2.2.2 The Minkowski Decomposition Operator

Minkowski decomposition is the restricted inverse of the Minkowski addition operation: it can be regarded as “erosion” or “erasing.”

Definition 4.2: Given two sets S and B in \mathbb{G}^d , *Minkowski decomposition* is the operation of determining a set T such that $S = B \oplus T$. According to our notation,

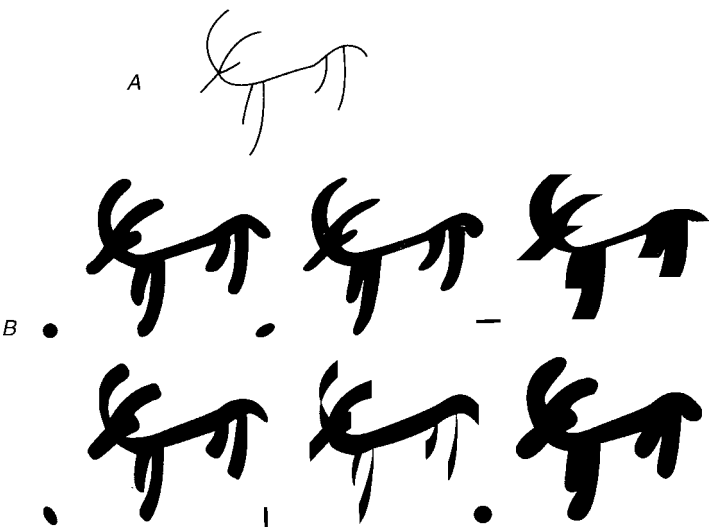


Figure 4.3 The members of the same equivalence class of shapes $A \oplus B$ (summand A is common to all of them)

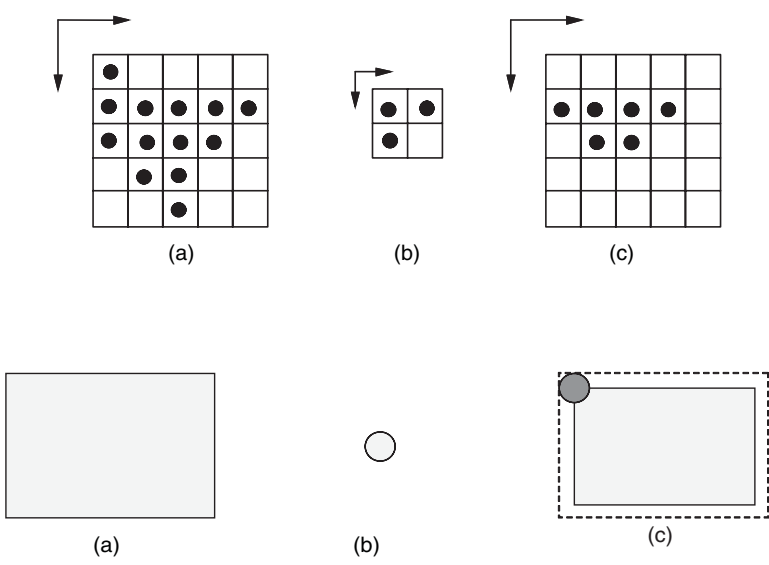


Figure 4.4 Minkowski decomposition in the discrete domain (top row) and in the continuous domain (bottom row): (a) the original pictorial image S ; (b) the structuring element B ; (c) their erosion $T = S \ominus B$

$$T = S \ominus B = \left\{ c \in \mathbb{G}^d \mid \forall b \in B, c + b \in S \right\}. \quad (4.3)$$

□

A simple example of Minkowski decomposition is shown below, in Figure 4.4. Conceptually, erosion (Minkowski decomposition) can be viewed as *erasing* – as a peeling away of the boundary much as we would peel off the layers of an onion.

Note that, in general, the decomposition $S \ominus B$ may not exist. For example, if S is a triangle, it cannot be expressed as a Minkowski sum of simpler shapes. This class of shapes is termed “indecomposable.” Even if S is decomposable, there may not be any T such that $S = B \oplus T$. For example, consider the case where S is a rectangle and B is a circle. In that case, we say that S is not decomposable by B .

However, later on we shall extend the definition of \ominus to ensure decomposition in every case.

We are now in a position to give you an intuitive idea of our proposed representation scheme. We demonstrate, in Figure 4.5, how a typical two-dimensional figure may be described by means of the scheme.

4.3 The Mathematical Nature of the Shape Operators

The first step in formalizing the model would be to explore (as completely as possible) the mathematical characteristics of the shape operators. It is then possible to use some “inference machine” (an idealistic concept, at present) to infer the required primitive shapes for describing a given set of shapes. Since the glue operator is a familiar one and in this chapter we are more interested in investigating the nature of Minkowski operators, we shall concentrate on the latter.

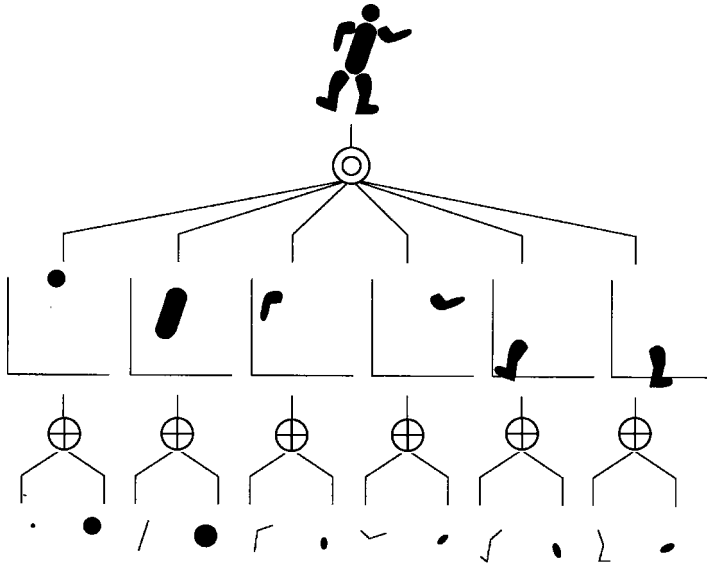


Figure 4.5 The description of a typical figure by means of the proposed scheme

4.3.1 The Minkowski Addition Operator

The obvious properties of the \oplus operator are as follows:

$$B \oplus T = T \oplus B, \quad (4.4)$$

$$(B \oplus T_1) \oplus T_2 = B \oplus (T_1 \oplus T_2), \quad (4.5)$$

$$\lambda(B \oplus T) = \lambda B \oplus \lambda T, \quad (4.6)$$

where λ is any real number. We also find that

$$B \oplus \{o\} = B, \quad (4.7)$$

where $\{o\}$ is the origin, and

$$B \oplus \emptyset = \emptyset. \quad (4.8)$$

We will state another important result, which is directly derivable from the definition of Minkowski addition by using a translational motion of a shape T by a point p ; that is,

$$T_p = T \oplus p = \text{Translate of } T \text{ by } p,$$

as shown in Figure 4.6, then it can also be shown in terms of set union or set intersection operations that

$$\begin{aligned} S &= \bigcup_{t \in T} B_t = \bigcup_{b \in B} T_b \\ &= \{s \mid (\check{B})_s \cap T \neq \emptyset\}, \end{aligned} \quad (4.9)$$

where

$$\check{B} = \{-b \mid b \in B\}$$

= symmetric set of B with respect to the origin,

and

$$(\check{B})_s = \text{Translate of } \check{B} \text{ by the translation } s.$$

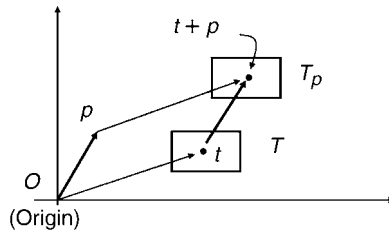


Figure 4.6 The translation of a figure T by a point p

Before we produce any more results, let us briefly note some of the implications of the above relations:

- (a) If $P(E)$ denotes the class of all the subsets of E (intuitively, the set of all shapes in E), then $(P(E), \oplus)$ forms an “Abelian semigroup.” That is, \oplus is a closed binary operation, which is associative and commutative (and therefore Abelian). In fact, it is more than a semigroup – it is a “monoid,” since there exists an identity element $\{o\}$ with respect to the \oplus operation. This classification has many advantages. It brings out the algebraic structure in the Minkowski addition operation, which may appear at first sight to be completely geometric. The properties of a semigroup, and particularly the associative and commutative properties of Minkowski addition, are frequently used in solving a number of practical problems (see the later part of this chapter). Also, the notion of “inverse shape” (briefly mentioned in the Conclusion of this chapter) has emerged in our attempt to extend the semigroup to a group.
- (b) Equation (4.9) is a “constructive” relation. It provides a method of computing the \oplus operation with the help of translation and the set union operation.
- (c) We can infer some of the geometric properties of the sum set $T \oplus B$. For example, if both T and B are convex, $T \oplus B$ is also convex. As a consequence, if the members of the set of primitive shapes K in our shape model are all convex and \oplus is the only shape operator, we cannot describe shapes that are nonconvex.

Also, if T , T' , and B are closed, bounded (i.e., compact), and convex subsets of E , then it can be shown that

$$\text{if } T \oplus B = T' \oplus B, \text{ then } T = T'.$$

Now, “uniqueness” is considered to be a preferred criterion in any shape description scheme. It states that any shape should be described in one and only one way within the model of the scheme. The relation shown above suggests that the uniqueness property could be achieved here to a considerable extent for compact, convex sets.

- (d) It is also possible to work out some of the topological properties of the product set $T \oplus B$. It is highly desirable that the shape operators in any shape description scheme will preserve the “topological validity” of the product shape. In other words, if our model for the objects to be described consists of compact (i.e., closed and bounded) sets in $\mathbb{G}^2/\mathbb{G}^3$, and if we start with primitive shapes that are all compact, the shape operators should be such that the product shapes will also be compact. For example, the set complement cannot be a possible shape operator in this case, since it allows unbounded objects from bounded primitives.

In the case of the \oplus operator, if both T and B are compact sets, $T \oplus B$ will also be a compact set. If T and B are closed sets (which may not be bounded), $T \oplus B$ also becomes closed. (This follows from (4.9), since the union of any finite family of closed sets is also closed.)

Another important topological property is “connectivity.” A set X is called connected if and only if any two points, x_1 and x_2 , in X can be joined by a curve that is completely included in X . In our case, it is easy to show that if both T and B are connected, then $T \oplus B$ will be also connected.

- (e) In practice, we often deal with shapes that are compact as well as convex. It is then interesting to note that the \oplus operation on compact, convex sets is very similar to the $+$ (addition) operation on positive, real numbers. This resemblance allows the user to form

a conceptually simple “mental schema” of a shape model using the addition operation. We can even define “divisibility” for the Minkowski sum \oplus . For any set B and any integer $n > 0$, we can obtain a set $(1/n)B$ that is positively homothetic to B . (The scalar multiple λA of a set A by a real number λ is said to be “homothetic” to A .)

Let us define S as

$$S = \overbrace{(1/n)B \oplus (1/n)B \oplus \cdots \oplus (1/n)B}^{n \text{ terms}}. \quad (4.10)$$

It can be proved that S converges toward $C(B)$ when $n \rightarrow \infty$, where “ $C(\cdot)$ ” denotes the convex hull. And if B is convex, then $S = B$. Therefore, B is “infinitely divisible” for \oplus if and only if it is convex – and not otherwise.

The set-theoretic formulation of the Minkowski addition operation allows us to devise many new results that turn out to be extremely useful for practical purposes, but that are difficult to foresee and prove by completely geometrical means. For example, let us note the following results:

$$T \oplus (B_1 \cup B_2) = (T \oplus B_1) \cup (T \oplus B_2), \quad (4.11)$$

$$T \oplus (B_1 \cap B_2) \subseteq (T \oplus B_1) \cap (T \oplus B_2), \quad (4.12)$$

$$C(B_1 \oplus B_2) = C(B_1) \oplus C(B_2). \quad (4.13)$$

The practical implications of these results are important. Let us mention a few of them below:

- (a) Equation (4.11) suggests that the addition can be done by taking the operands piece by piece and then combining the intermediary results by unions. We are now, for example, provided with a technique for computing the Minkowski sum of complex shapes using much simpler operations. Any complex nonconvex shape can be expressed as unions of convex shapes. Therefore, just an algorithm for the addition of two convex bodies and an algorithm for the union of two shapes are sufficient to carry out the addition of complex objects.
- (b) Let P and Q be two convex polytopes (a polytope, the analogue of a polygon in two dimensions and a polyhedron in three dimensions, is a figure bounded by hyperplanes in d dimensions) and let $\text{vert}(P)$ and $\text{vert}(Q)$ be their respective vertices. This means that $P = C(\text{vert}(P))$ and $Q = C(\text{vert}(Q))$, where “ C ” denotes the convex hull. Now, from (4.13) we can write

$$P \oplus Q = C(\text{vert}(P) \oplus \text{vert}(Q)). \quad (4.14)$$

The above equation provides a very simple but elegant method of computing two convex polytopes. Add the vertices of the operands and take their convex hull. You will obtain the sum polytope.

4.3.2 The Minkowski Decomposition Operator

When we are dealing with Minkowski decomposition, the first question that strikes us is as follows: “Can we, as in the addition operation, define decomposition too in terms of set

operations?” Equation (4.9) provides a clue to defining decomposition in that way:

$$S \ominus B = \bigcap_{b \in B} S_{-b} = \{t \mid B_t \subset S\}. \quad (4.15)$$

But we must note a discrepancy between the definition of decomposition given in Section 4.2 and the above definition. In Section 4.2, we stated that, in general, the decomposition $S \ominus B$ may not exist. On the other hand, (4.15) always ensures a realizable $S \ominus B$ value. The fact is that (4.15) is a more general definition than that given in Section 4.2. Equation (4.15) ensures that if S is decomposable by B – that is, if there exists a set T such that $S = T \oplus B$ – then $(S \ominus B) \oplus B = S$, which conforms to the definition in Section 4.2. For all future purposes, we shall follow the general definition of decomposition expressed by (4.15).

Equation (4.15) can also be expressed in the form

$$S \ominus B = (S^c \oplus \check{B})^c, \quad (4.16)$$

where the superscript c denotes the complement. Note that

$$B \ominus B \supseteq \{o\}. \quad (4.17)$$

If and only if B becomes a bounded set, we have the equality

$$B \ominus B = \{o\}. \quad (4.18)$$

Some aspects of the physical significance of these results can be noted:

- (a) Equation (4.15) provides a constructive method for computing the \ominus operation using translation and set intersection operations.
- (b) If $S = T_1 \oplus B = T_2 \oplus B = \dots = T_i \oplus B = \dots = T_n \oplus B$, then according to the definition of (4.15), $(S \ominus B)$ will produce the biggest set, say, T_m of all those T_i 's. To be more precise, $T_m = T_1 \cup T_2 \cup \dots \cup T_n$, and $S = T_m \oplus B$.
- (c) If S is convex, then $(S \ominus B)$ will be also convex, irrespective of the shape of B .
- (d) Some of the topological properties of the \ominus operation can also be inferred. If both S and B are compact sets, $S \ominus B$ must be a compact set. Moreover, if S and B are both closed, $S \ominus B$ is also closed (since intersections of closed sets are closed).

However, on the question of connectivity, the behavior of the \ominus operation is different from that of the \oplus operation. Even if both S and B are connected, $S \ominus B$ may not always be connected.

The importance of boundedness in the case of the \ominus operation must be noted by examining (4.17) and (4.18). The conceptualization of the implication of (4.17) is not obvious. In practice, however, we are mostly concerned with compact sets where (4.18) holds true.

- (e) We hope that you have already noted some resemblances and differences between Minkowski addition and decomposition of shapes and arithmetic addition and subtraction of real numbers. For example, like arithmetic addition, Minkowski addition is also commutative and associative, and an identity element also exists. However, they differ on the question of their inverse.

We will discuss this resemblance more intensively and extend our notions of Minkowski addition and decomposition in the next chapter. Here, we will note some more resemblances, without any detailed discussion.

Some of those resemblances may also be found when we examine the “properties of order.” In real number arithmetic, the order is customarily expressed in terms of a relation $<$, called “less than,” which is of course well known. Let us examine the order properties in the case of Minkowski addition and decomposition of shapes.

$B \subseteq C$ implies $T \oplus B \subseteq T \oplus C$, and

$$S \ominus B \supseteq S \ominus C,$$

$$B \ominus S \subseteq C \ominus S.$$

Therefore, the mental schema of the arithmetic of real numbers with addition and subtraction operators closely resembles the arithmetic of shapes with Minkowski operators, as far as the question of properties of order is concerned.

The resemblance can be extended further by showing the following relationship:

$$(S \ominus B_1) \ominus B_2 = (S \ominus B_2) \ominus B_1 = S \ominus (B_1 \oplus B_2). \quad (4.19)$$

However, there are also many differences. For example, if $B \subseteq C$, the set $B \ominus C$ is not always empty as we might expect. $B \ominus C$ becomes an empty set when B and C are compact sets.

We can see the difference even in the relationship

$$(S \ominus B_1) \oplus B_2 \subseteq (S \oplus B_2) \ominus B_1. \quad (4.20)$$

However, equality holds if the sets are compact, convex sets.

In a similar manner, we can show some further resemblances and differences. But there are some more fundamental questions: “Why do these resemblances and differences arise?” “Is it possible to bring about any uniformity between these two fields?” We expect to be able to answer these questions in the near future.

We also obtain the following formulae:

$$S \ominus (B_1 \cup B_2) = (S \ominus B_1) \cap (S \ominus B_2), \quad (4.21)$$

$$(S_1 \cap S_2) \ominus B = (S_1 \ominus B) \cap (S_2 \ominus B), \quad (4.22)$$

$$S \ominus (B_1 \cap B_2) \supseteq (S \ominus B_1) \cup (S \ominus B_2), \quad (4.23)$$

$$(S_1 \cup S_2) \ominus B \supseteq (S_1 \ominus B) \cup (S_2 \ominus B). \quad (4.24)$$

The usefulness of these set-theoretic results can be made more apparent in the context of applications. To clarify our point, let us mention a few of the implications of these results:

- (a) Equation (4.21) suggests that the \ominus operation could be performed by taking set B piece-by-piece and then combining the intermediary results by intersection.
- (b) We present below an example where the decomposition of nonconvex bodies (a difficult task) can be converted into the decomposition of convex bodies (a relatively simpler task) by applying (4.22).

Equation (4.22) allows some S to be broken down as $S = S_1 \cap S_2$, so that we can work on S_1 and S_2 separately. Let us consider an example.

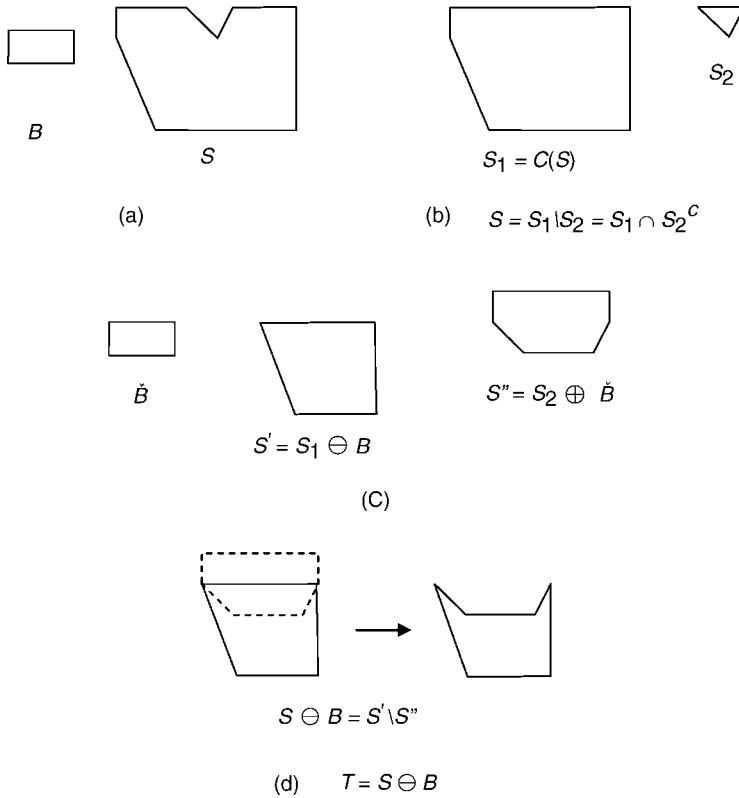


Figure 4.7 A nonconvex sum and a center-symmetric convex summand: (a) the original S and B ; (b) an interpretation $S = S_1 \setminus S_2 = S_1 \cap S_2^c$; (c) the convex sum and decomposition; (d) the resulting nonconvex sum $T = S \oplus B$

Let the given S be a nonconvex polygon and let B be a center-symmetric convex figure, as shown in Figure 4.7. We can express S as the intersection of two convex parts; that is,

$$\begin{aligned}
 S &= C(S) \setminus (\text{hole}) \quad [“\setminus” \text{ denotes set difference}] \\
 &= S_1 \setminus S_2 \\
 &= S_1 \cap S_2^c,
 \end{aligned}$$

where $C(\cdot)$ denotes the convex hull. Therefore,

$$\begin{aligned}
 S \oplus B &= (S_1 \cap S_2^c) \oplus B \\
 &= (S_1 \oplus B) \cap (S_2^c \oplus B) \\
 &= (S_1 \oplus B) \cap (S_2 \oplus \check{B})^c \\
 &= (S_1 \oplus B) \setminus (S_2 \oplus \check{B}).
 \end{aligned}$$

The above equation shows that we need to operate only on convex polygons (provided that B is convex) in order to achieve the required decomposition. The result of this operation is shown in Figures 4.7(b), (c), and (d).

It is certainly not possible here to note all of the results that are available so far on Minkowski operations: we have selected only those that are relevant for our future discussions in this chapter. But it must be mentioned that we are leaving out a set of properties that seem to be useful in connection with shape description. We know that, apart from purely geometrical and topological properties, there are also other shape properties, such as the surface area, volume, or centroid of a shape. A inquiry of the following kind is quite natural within the framework of a shape model. What will the area of the sum polygon $T \oplus B$ be if the areas of polygons T and B are $\text{Area}(T)$ and $\text{Area}(B)$, respectively? Readers interested in these kinds of shape properties are invited to refer to Lyusternik [62] or Matheron [69].

4.4 A Few Reasons for Choosing Minkowski Operators as Shape Operators

The set union and intersection operators are the most widely used shape operators in the existing shape description schemes. The sweep representation scheme is one of the notable exceptions, though restricted to the limited domain of representing shapes that have either translational or rotational symmetry. Unfortunately, the potential of Minkowski operators as shape operators has hardly been explored, though they seem to be “natural description tools” for multifarious uses of computers in the design, manipulation, analysis, and modeling of physical, manufactured, and biological shapes. Some of the reasons for choosing them as shape operators for our model are mentioned below. More justifications can be found in the next section, where we will briefly describe some of the areas of application.

4.4.1 A Natural Description Tool

It has already been argued that, in order to minimize the computational effort during interactions with a shape model, the internal description of the shape should be as close as possible to the mental schema of the user. How good is our shape model in this regard?

Let us consider the addition operation. It is like sweeping a brush or a cutter tool along a path – like “painting” in two or three dimensions. It is like adding flesh to a skeleton. Similarly, it is like moving an object through space. It might even be thought of as the “inflation” of the boundary of an object, just as we would inflate a balloon.

The decomposition operation is the inverse operation. Therefore, it is like “erasing” an area/volume with a two- or three- dimensional eraser. Or it might be viewed as the “peeling away” of the boundary, much as we would peel off the layers of an onion.

Because of the simple and natural mental schema, these operations become excellent abstractions of a number of tasks in a variety of areas of application, such as geometric modeling, the description of biological forms, spatial planning, the containment of polygons, graphic arts and digital typesetting, image processing, and so many others. The Minkowski operators turn out to be a very important tool in solving a number of geometric problems, and can be considered as an important addition to computational geometry.

4.4.2 The Large Domain of the Model

A quick impression of the domain of the shape model can be obtained from the following three theorems.

Theorem 4.1 (the triangle theorem). *With a triangle and a line segment (degenerated triangle) as the primitive shapes in two-dimensional space and Minkowski addition as the shape operator, we can describe all convex polygons.*

Theorem 4.2 (central symmetry). *With a line segment as the only primitive shape and Minkowski addition as the shape operator, every “zonotope” in d dimensions can be described.*

Theorem 4.3 (translational symmetry). *Any shape having translational symmetry can be expressed as the Minkowski sum of two simpler shapes.*

The proofs can be found in Yaglom [99], Grünbaum [34], Matheron [69], and Ghosh [26].

Let us elaborate the implications of these theorems a little further to give a better physical picture of the domain. As an example, zonotopes, which are convex and centrally symmetric polytopes, are very commonly encountered in practice. Cubes and rectangular parallelepipeds are zonotopes in three dimensions. All centrally symmetric convex polygons in two dimensions – that is, squares, parallelograms, regular hexagons, and so on – are also zonotopes. Even a right circular cylinder can be approximated arbitrarily closely as a zonotope, as can, similarly, a circle or an ellipse. In more general terms, every convex polyhedron bounded by parallel-sided $2m$ -gons is a zonotope. Theorem 4.2 states that all such shapes can be generated by the addition of straight-line segments only.

Theorem 4.3 is concerned with translationally symmetric objects. We have plenty of these objects around us in the real world. Cylinders, prisms, rectangular boxes, triangular wedges, circular rings, and two-dimensional ribbons are all translationally symmetric objects. Even many biological objects, including the human body, exhibit translational symmetries at some level of detail. According to Theorem 4.3, these objects can be described as Minkowski sums of much simpler objects. For example, a cylinder is a Minkowski sum of a circular disk and a straight-line segment in three-dimensional space. For animation purposes, it is advisable to model the human body as a “stick-figure skeleton.” The flesh in the body can be built up by adding circular disks of various sizes to the skeleton (see Figure 4.3).

The implications of Theorem 4.1 are tremendous, and must be investigated further. The theorem implies that any convex two-dimensional shape can be approximated arbitrarily closely using only triangles, line segments, and the Minkowski addition operator. In other words, the theorem states that any convex polygon can be decomposed into triangles and line segments. Obviously, this provides a new way of characterizing convex polygons, and essentially all convex shapes in two dimensions. For this very reason, the set of all triangles and line segments forms a class that is called the “universal approximating class in R^2 ” with respect to Minkowski addition.

Surely it is natural to ask here whether any universal approximating class exists in R^3 for convex bodies. Can the set of all simplexes – that is, line segments, triangles, and tetrahedra – describe all convex polyhedra using the Minkowski addition operator as the shape operator?

Unfortunately, this is not the case. There is a class of polytopes, known as “indecomposable polytopes,” that cannot be expressed as the Minkowski sum of simpler shapes. Consider the following theorem.

Theorem 4.4 (the 2-face theorem). *If all the 2-faces (two-dimensional faces) of a convex polytope S are triangles, then S is indecomposable.*

This means that tetrahedrons and octahedrons are indecomposable.

Note that even if a polytope is decomposable, it may not be expressed as a Minkowski sum of simplexes. For instance, in three dimensions, the Minkowski sum S of a square pyramid and a line segment parallel to an edge of the base of the pyramid cannot be expressed as a Minkowski sum of tetrahedra.

Even though the set of simplexes and Minkowski addition cannot describe all of the convex shapes, they can nevertheless describe a very large class of convex shapes.

Is the shape model improved through the introduction of the decomposition operator? The introduction of the “inverse” (i.e., “undo”) operation can be justified in terms of the “expressive completeness” of a system. But, in particular, how does it help in extending the description domain?

Although the decomposition operation cannot generate any nonconvex shapes if the primitive shapes are all convex (see implication c of Minkowski decomposition in Section 4.3), it can considerably increase the convex domain. (The \ominus operator has the capability of doing something like a “similarity transformation” – a very interesting property that will be discussed in a later section.) The Minkowski decomposition operator plays a particularly significant role in describing nonconvex shapes. Without going into the mathematical rigor, we can demonstrate this by means of the example shown in Figure 4.8. Note that the nonconvexity in the resulting shape T' has been increased compared to that in the operand shapes S and B .

The third shape operator in our model is the glue operator \odot , which has been included for the following reasons. The glue operation is a very common phenomenon in practice, and it is also very easy to conceptualize the operation. Moreover, we have yet to find out whether the glue operation can be expressed in terms of Minkowski operations.

Our model, with the triangle and the line segment as primitive shapes and the addition and glue operators as the shape operators, can now approximate any two-dimensional shape (both convex as well as nonconvex) arbitrarily closely. Similarly, the glue operator can be effectively used in three dimensions to describe indecomposable and complex nonconvex shapes. (Our future aim is to use the Minkowski operators alone to describe most of the frequently occurring geometric shapes, by carefully choosing a small set of primitive shapes.)

To give an impression of the rich domain of our model, we will present a few examples (Figure 4.9), in which complex shapes are represented by using these shape operators.

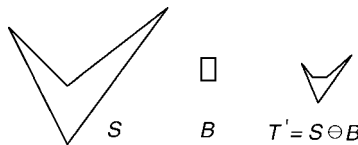


Figure 4.8 Nonconvexity of shape and the role of the decomposition operator

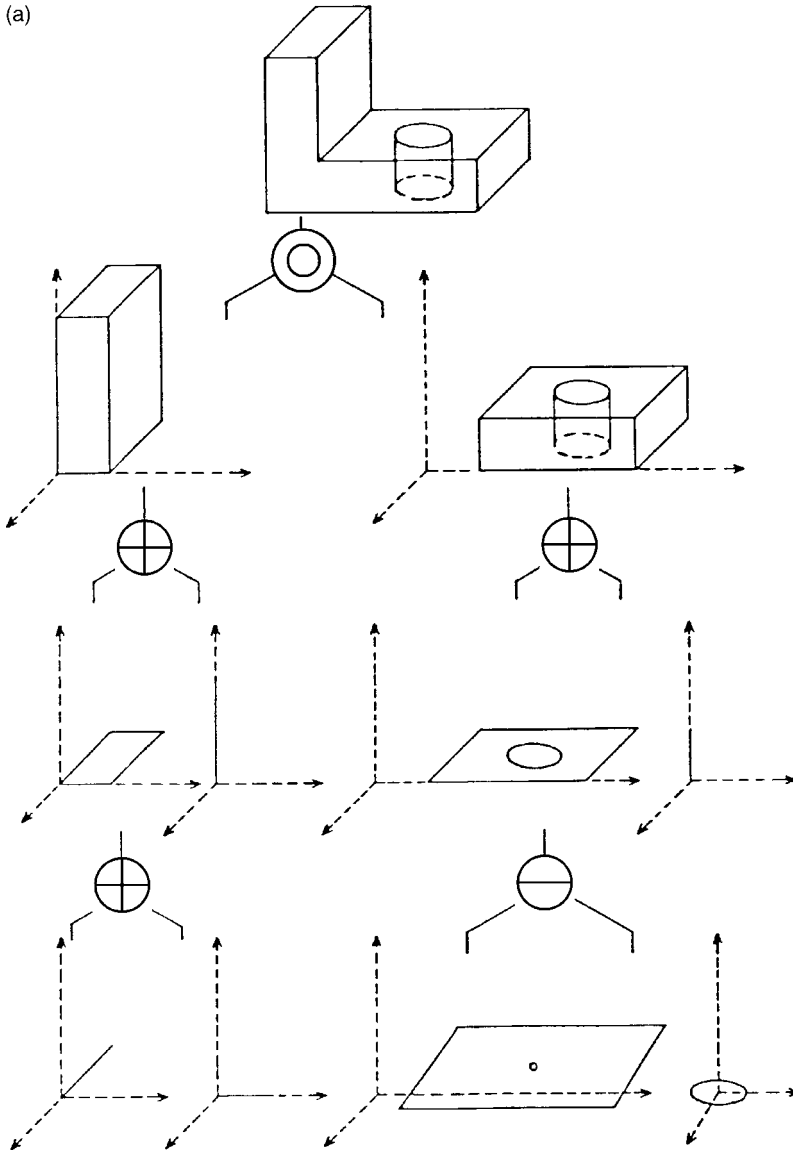


Figure 4.9 A demonstration of the domain of the shape model: (a) the representation of a three-dimensional shape using Minkowski addition and decomposition and the glue operator; (b) the generation of three-dimensional shapes using the \oplus and \odot operators; (c) a representation of the union of two cylinders

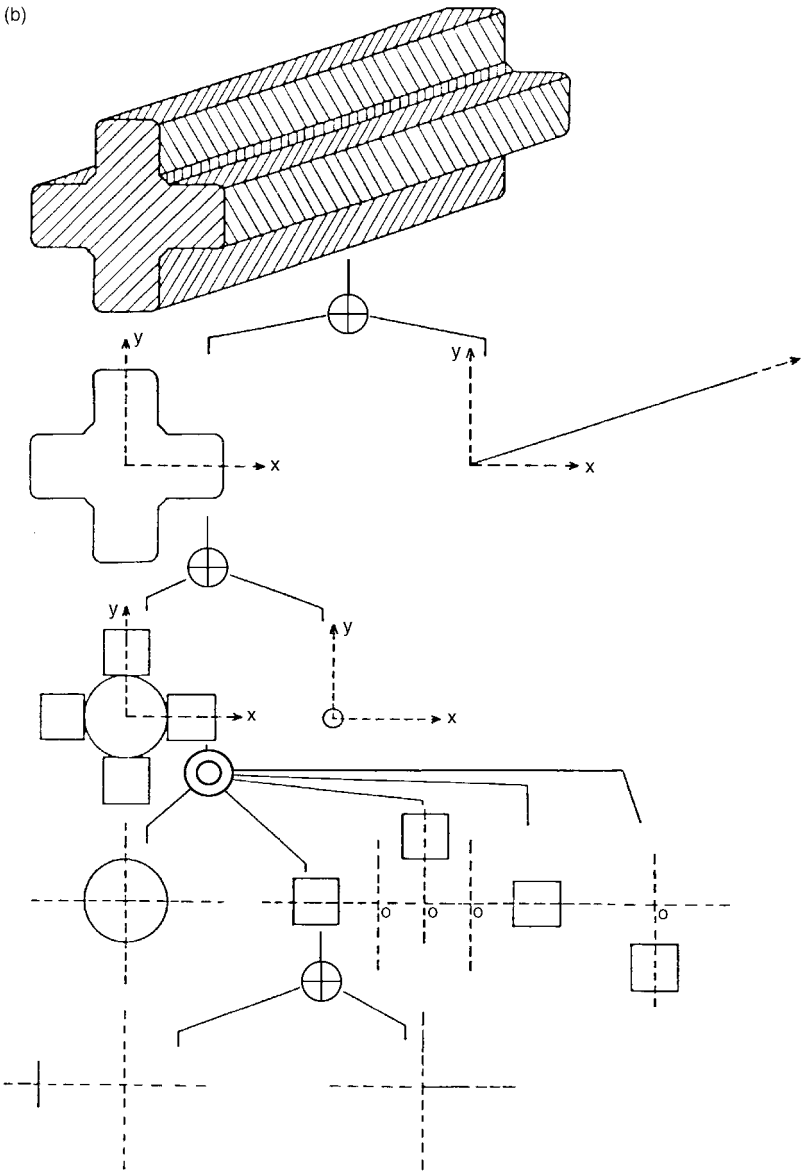


Figure 4.9 (Continued)

4.4.3 Conciseness in Shape Representation

“Conciseness” refers to the size of the representations in a scheme. Obviously, our model is very concise from representational point of view. It is often possible to represent three-dimensional solids in terms of two-dimensional regions, or even only using one-dimensional lines. It may be possible to represent a polyhedron with total number of vertices $M \cdot N$ as the sum of two

(c)

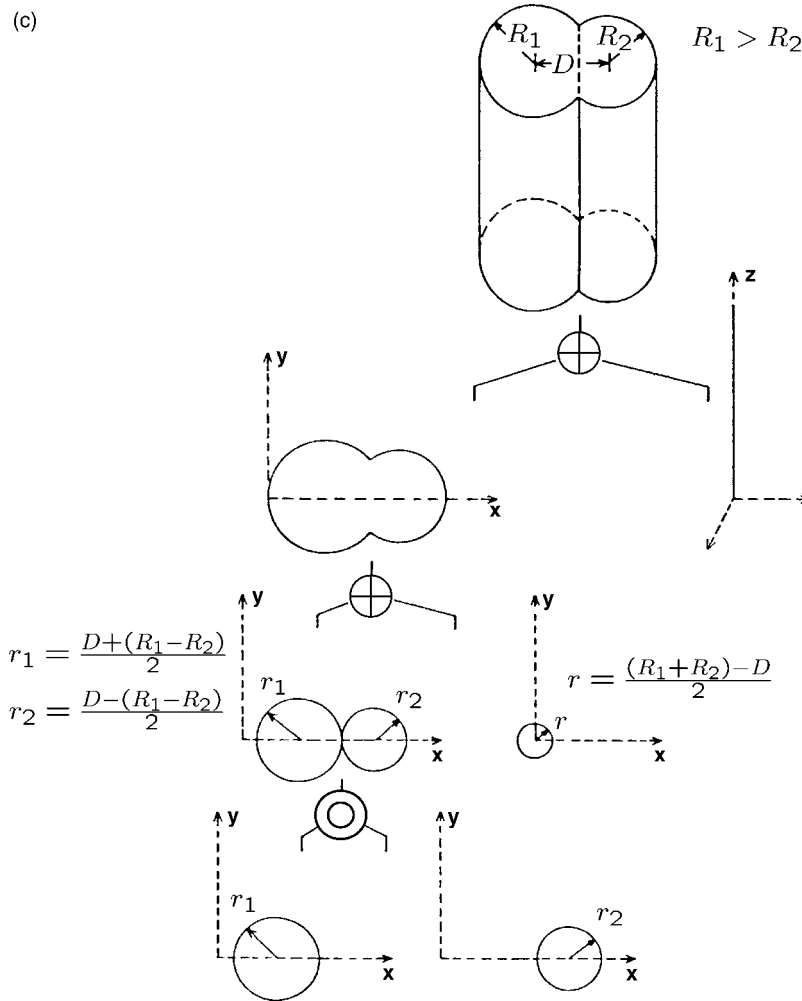


Figure 4.9 (Continued)

polyhedra with M and N vertices, respectively. Moreover, complex shapes such as shown in Figure 4.10 can be represented reasonably concisely using very simple primitives. To elaborate this point, some further examples are provided in the next section.

The conciseness is achieved due to the fact that Minkowski addition in R^n is not a function in R^n , but it is a function in R^{2n} .

4.4.4 The Geometric Nature of the Shape Operators

Any mathematical model is an abstract concept. The extent to which this abstract concept can be usefully regarded as a description of the shape of an object that exists in the real world (or that potentially exists in the real world) depends on the extent to which that characteristics

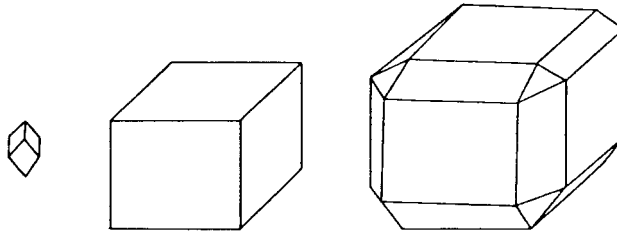


Figure 4.10 A concise representation of a complex object

of the object “correspond” to the characteristics of this abstract concept. Thus, the application of any mathematical model of a shape description always singles out certain characteristics of an object and suppresses others. It is often the case that we are primarily interested in the “geometric” and “topological” aspects of objects, rather than the mass properties or other characteristics. Therefore, a shape model and, in particular, the shape operators have to be examined in this context.

Do the set union, intersection, and difference operators, which are frequently chosen as shape operators, serve this purpose well? We must remember that a shape in real Euclidean space is much more than a mere set of points. This set of points has a certain structure associated with it; for example, these points have neighborhoods, these neighborhoods are related to intervals, and so on.

Real Euclidean space is a “topological space” as well as a “metric space” (R^n, d) , where d denotes “distance,” which is the metric on R^n . However, the set operations do not take this structural concept into account at all, since they are very general and applicable to any set. This “generality” of set operators may pose problems in our intuitive understanding of shapes and the interactions between them. For example, an object (A) with a hole in it is usually expressed as the difference of the solid object (B) and the hole (C); that is, $A = B \setminus C$. Intuitively, it is assumed that C is a subset of B . Unfortunately, this implicit assumption cannot be made internal to the set operation $B \setminus C$. An operation such as $[(\text{Object } 1 \cup \text{Object } 2) \setminus \text{Object } 2]$ may generate Object 1 at one instant and a null object at some other instant. It is necessary to check the physical validity of objects produced solely by machine using the set operators. When two objects touch each other, intersection or difference operations often cause problems.

The point we want to stress is that the concept of the geometric and topological structure of real Euclidean space should somehow be embedded in the nature of the shape operators that we choose. It is then possible to eliminate the types of problems that we mentioned earlier. And, to this extent, the Minkowski operators are the right candidates, since these operations are defined on vectors in real Euclidean space.

4.5 Geometric Modeling by Minkowski Operations

4.5.1 Better Shape Representation

Geometric modeling has been becoming increasingly important in the field of CAD/CAM. The aim of geometric modeling is to produce representations of real objects – representations that permit (at least in principle) any well-defined geometrical and topological property of

any represented object to be calculated automatically. Therefore, all we need for geometric modeling is an unambiguous mathematical model for object representation, and also a number of geometric algorithms to manipulate that model in order to carry out various operations on the objects.

The shape model that we have presented in the previous sections of this chapter is a model for both representation and analysis of the shapes of objects. Therefore, it can certainly be used for geometric modeling purposes. What we have to do is to choose a set of “suitable” primitive shapes and then use our shape grammar to represent the complex objects. Of course, choosing a set of suitable primitive shapes depends on the class of objects to be represented, on the type of operations to be performed on the objects, and so on. Therefore, it turns out to be an extremely difficult task, for which no formal guidelines are available (for further details, see Requicha [84], Arbab [2], and Ghosh [26]).

The most pertinent questions here are as follows: “Is our model better than the other existing shape representation models?” “Do the Minkowski operators offer any extra advantage in representation over the others?” It is important to note that no representation scheme is better than all others for all purposes. However, for estimating the overall quality of a representation scheme, Requicha [84], Gardan [23], and so on have defined a set of criteria that the scheme should satisfy, some of which are as follows: a large representation domain, a high level of conciseness, physical validity of the syntactically correct representation, unambiguousness in representation, and ease in creation and modification of the represented objects.

Our model can also be assessed in terms of these criteria, and we consider that most of them are satisfied. The domain and the conciseness of the model have been already discussed in Section 4.4. It has also been mentioned in Section 4.3 that if the primitive shapes chosen by us are all valid and unambiguous, then the nature of the shape operators ensures the validity and unambiguousness of the complex shapes generated by our shape grammar. Ease in the creation and modification of shapes depends primarily on two factors – the representation should be concise and it should also be as close as possible to the mental schema of the user. These two aspects of our model have also been discussed in Section 4.4. One of the desirable criteria is uniqueness. Unfortunately, like most of the other shape representation schemes, our model too does not guarantee uniqueness in representation. The same object can be represented in several ways: for example, a centrally symmetric convex hexagon can be generated by adding two triangles, as well as by adding three straight-line segments. Uniqueness may be achieved by properly choosing the set of primitive shapes.

However, the actual superiority of our model over the others should be judged in the context of representation of objects that are human-made. Almost everywhere in the domain of human-made shapes, we find in action the “motion of a tool” – from the writer’s pencil, the calligrapher’s pen, the artist’s brush, and the sculptor’s chisel to the machine’s drill-bit. And the Minkowski operators turn out to be an excellent abstraction to capture the translational motion of a tool. Let us clarify this point further.

4.5.2 A Procedural Model

Our representation model might be termed “procedural,” since it defines a shape by describing a method for its production – by painting and/or erasing. In contrast to this, almost all other schemes (except sweep representation) define a shape as it is seen. Our method of representation is somewhat analogous to the representation of a series of numbers by defining

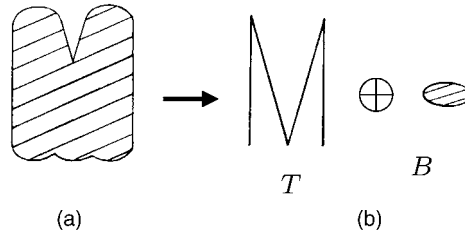


Figure 4.11 A complex two-dimensional figure and its B - T representation

a generating function, instead of writing down each and every number of the series. Therefore, it is often possible to represent a complex shape much more easily and concisely through our scheme, while boundary or other representations of that shape may become quite cumbersome. For example, consider the two-dimensional figure shown in Figure 4.11(a). Its boundary representation is difficult, while representing it as an elliptical brush B moved over a polyline trajectory T (as shown in Figure 4.11(b)) is certainly simpler. Procedural models turn out to be particularly useful in three dimensions. They can be used very effectively in the context of manufacturing automation, since the construction of a solid volume can be naturally described by means of a cutter (B) placed at every point of some trajectory (T).

4.5.3 The Internal Structure of a Model

Most of the existing shape representation schemes aim to capture the external aspects of the shape that an object presents to the outside world. Our model, on the other hand, represents the shape by defining how the shape is formed. Thus it provides information about a kind of internal structure of the shape, which cannot be perceived from the mere external aspects. This point is important and was discussed in detail in Ghosh [26]. Here, we will briefly mention a few examples to clarify our point:

1. Let us refer to Figure 4.11 again. It is not difficult to perceive that the elliptical brush region B , when moved over the M-shape trajectory curve T , will produce the original two-dimensional figure shown in part (a). But it is not so immediate to realize that the same original figure can be generated by reversing the role of the brush and the trajectory. That is, in this case we can also consider the M-shape curve as the brush and the elliptical region as the trajectory, since $B \oplus T = T \oplus B$. Conceptualizing a region or a volume as a trajectory does not come spontaneously.
2. It has already been mentioned in Section 4.4 (see Theorem 4.1) that any convex polygon can be represented as a Minkowski sum of triangles (since straight-line segments can be considered as degenerate triangles). These constituent triangles capture the internal structure of the polygon in the same way that the constituent atoms capture the internal structure of a molecule. But these triangles are not immediately identifiable from the external shape of the polygon; they are more latent, more internal. We must employ separate computational tools to identify them and separate them out. This is demonstrated in Figure 4.12, where a convex polygon and its constituent triangles are shown.

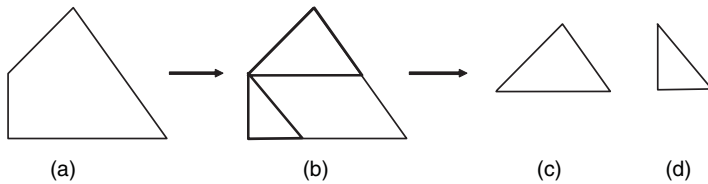


Figure 4.12 A typical convex polygon and its constituent triangles: (a) a convex polygon; (b) its decomposition; (c) the constituent triangles

- Our shape model provides us with the criteria to classify or categorize shapes in terms of their representations. One immediate classification is the indecomposable shapes (such as the triangle or the tetrahedron) and the decomposable shapes (such as the square or the cylinder). The decomposable shapes can again be subdivided into different classes: such as zonotopes, which are decomposable into straight-line segments; two-dimensional convex polygons, which are decomposable into straight-line segments and triangles; a class of convex polyhedra, which are decomposable into straight-line segments, triangles, and tetrahedra; and so on. More interestingly, the convex and nonconvex categorization of shapes can also be applied within our framework. Note that the product shape $A \oplus B$ or $A \ominus B$ will be always convex if the operands are convex, while for nonconvex operands the product may be convex or nonconvex. Also, a set is “infinitely divisible” for \oplus if and only if it is convex, and not otherwise (see Section 4.3). Moreover, if $A \oplus B = A' \oplus B$, then $A = A'$ if A , A' , and B are all convex, compact sets. There are more such results for convex shapes.

4.5.4 Concise Representation

In the previous section, we have already discussed the fact that our model is highly concise from a representational point of view. We emphasize this point again by citing the following example.

The “spatial occupancy” scheme is one of the most popular shape representation schemes at the present time. The representation is essentially a list of spatial cells occupied by a three-dimensional solid or a two-dimensional region. The most commonly used spatial cells are a square in two dimensions (known as a “pixel”) and a cube in three dimensions (called a “voxel”). It is easy to see that the spatial occupancy scheme is, in some sense, equivalent to our model. Referring to Figure 4.13, we can assume that one of the summands is a discrete-valued operand – that is, a set of points – while the other summand is the spatial cell in use. In fact, we can go one step further. The commonly used spatial cells in two dimensions are the square, the rectangle, and the regular hexagon. All of these cells can be represented as Minkowski sums of straight-line segments only. In three dimensions, there is a special class of polyhedra that consists of five “primary parallelohedra,” which are known as “space-fillers.” Since the space-fillers, when stacked, are able to completely fill the space, any one of them can be used as a spatial cell in the spatial occupancy scheme. These five space-fillers are the cube (Figure 4.14(a)), the hexagonal prism (Figure 4.14(b)), the rhombic dodecahedron (Figure 4.14(c)), the elongated dodecahedron (Figure 4.14(d)), and the tetrakaidecahedron

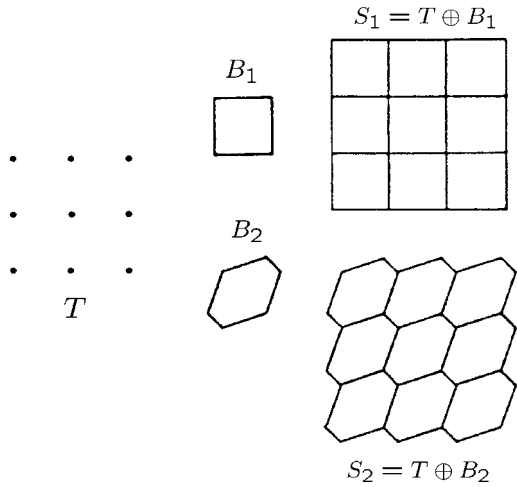


Figure 4.13 Spatial occupancy representation as a Minkowski sum

(Figure 4.14(e)). The cube and the hexagonal prism do not need any further explanation. The rhombic dodecahedron is bounded by 12 equal rhombic faces. The elongated dodecahedron has eight equal rhombuses and four equilateral hexagons. The tetrakaidecahedron is bounded by six squares and eight hexagons.

The interesting point is that all of these five space-fillers are zonohedra; they can be represented as the Minkowski sum of straight-line segments. We can, therefore, summarize our discussion by noting that, even in three dimensions, only the point, the straight-line segment, and the \oplus operator are needed to simulate spatial occupancy representation – and, therefore, the amount of conciseness in representation should be self-evident.

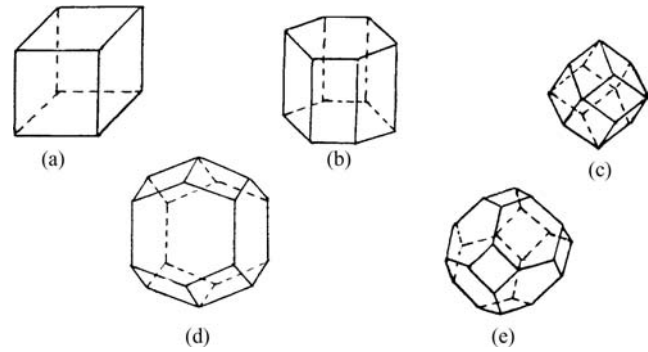


Figure 4.14 The space-fillers in three dimensions

4.6 Image Analysis by Minkowski Operations

4.6.1 Mathematical Morphology

So far, we have investigated the nature of the Minkowski operators as shape operators. In this section, our primary purpose is to demonstrate that a number of practical problems from seemingly different fields can be reformulated in terms of Minkowski operations. Thus we find that there is an excellent possibility of unifying a number of areas that, from a conventional viewpoint, appear to be completely separated from each other. Since it is not possible to cover all such application domains in this section, we will briefly outline only the representative ones.

In the field of image analysis, the typical set-theoretic approach is called *mathematical morphology* [90]. Mathematical morphology consists of a class of shape operators that are derived from Minkowski addition and decomposition. Note that those operators have special names in morphology, such as dilation for Minkowski addition, erosion for Minkowski decomposition, and so on. In this section, we will use a mixture of both terms, as long as this will not cause any confusion.

Mathematical morphology is now a well-established method. While the final goal of most image processing methods is to segment images into meaningful units and textures in accordance with the judgment of the human eye, the aim of the mathematical morphology method is to define the structure of an object by the set of relationships that exists between the various parts of the object. Therefore, the theory of mathematical morphology views images from the perspective of geometrical structure, thus distinguishing itself from other image processing theories; for example, syntactic theories based upon generative grammars, or signal processing theories based upon Fourier analysis. In simple terms, the idea of mathematical morphology is as follows. Let the image that has to be processed be a set of points, X , in real Euclidean space. In mathematical morphology, a number of operations are defined for modifying the shape of the image X by another shape, say B , which is known as the “structuring element.” The commonly used morphological operations are the dilation and erosion, and the opening and closing, of X by B . It is the task of the morphologist to choose a structuring element B carefully. The sphere is regarded as a universal structuring element, because of its perfect symmetry. Other frequently chosen structuring elements are the circle and square in two dimensions, and the cone, cylinder, and paraboloid in three dimensions. Once a B is chosen, the morphologist modifies the shape of X by B according to the morphological operations and reduces X to a sort of caricature, which is more expressive than the actual initial image X .

In 1964, a research institute was opened by J. Serra and G. Matheron, on the campus of École des Mines of Fontainebleau (France), in order to shed light on the relation between the geometric features of an ore and its physical properties. They began to develop a method with which to analyze the textural pattern that appears in the cutting plane of rock. This was the beginning of mathematical morphology.

Originally, morphology was the study of the relationships between the nature of cell structures in biology, organs, or mineral materials in rock and the resulting macroscopic appearance. As the name suggests, the mathematical morphology of Serra *et al.* analyzes various structures of patterns mathematically. The greatest feature is to develop a method of analyzing the global structure of a pattern only by using the local information on their input patterns. The same approach is also seen in biological morphology, the typical example being attempts to explain the generation of a living organism through the interactions between adjoining cells. However,

in this case, not only the local view but also the global view is commonly taken into account, as the textural pattern that minimizes the free energy of the whole system is the most stable.

Another feature of mathematical morphology is that the objective is not necessarily to produce a geometric structure or a texture that can exist in itself, but one that is materialized in correlation with the object and the observer. The concept of the structuring element was introduced on this basis. Although a structuring element is a certain simple pattern, the interaction of this element with an object results in an embellished or symbolic form of the object. By considering structuring elements, in fact, the technique became powerful. But, on the other hand, the choice of a suitable structuring element has always remained one of the main problems, as we shall see later.

Since mathematical morphology is an image-analysis approach with simple operation using local information, it is suitable for the construction of a machine process, especially one for parallel processing. Several studies have been published in which conventional image processing techniques are replaced by morphological operations (for details, see [89, 67, 68, 69, 38]).

4.6.2 Morphological Operators

The fundamental problem of mathematical morphology was to define the morphological operations in such a way that they could bring out the geometric structures of an image very expressively, and the morphologists eventually discovered that the Minkowski addition and decomposition operations are the best choices for that task. Therefore, we find that the Minkowski operators are present at the core of all morphological operations, a few of which – expressed in terms of Minkowski and set operations – are as follows:

- A1. Dilation of X by B : $= X \oplus B$.
- A2. Erosion of X by B : $= X \ominus B$.
- B1. Opening of X by B : $= (X \ominus B) \oplus B$.
- B2. Closing of X by B : $= (X \oplus B) \ominus B$.
- C1. Thinning of X by B : $= X \cap [(X \ominus B_1) \cap (X^c \ominus B_2)]^c$.
- C2. Thickening of X by B : $= X \cup [(X^c \ominus B_1) \cap (X \ominus B_2)]$.

In the above, $B = \{B_1, B_2\}$.

Two important questions can be asked here: “What are the geometrical interpretations of these morphological operations?” “In other words, what kinds of geometrical structures do they bring out from the images?” These and some other related questions have been already answered by the morphologists – see the literature cited above.

Let us consider two combinations of dilation and erosion. One is that erosion is applied first, and then dilation is applied to produce the result. The other is the opposite: that dilation is applied first, and then erosion is applied to produce the result. In both cases, the result is that a portion smaller than the structuring element is removed from the original shape. The former operation is called “opening” and the latter is called “closing.”

For example, opening for which a circle is used as a structuring element makes convex vertexes round and removes thin capes and small islands. Closing for which a circle is used as a structuring element makes sharp dents round and fills thin cracks, long and thin bays, and small holes.

We can define opening and closing formally.

Definition 4.3 (opening): The *opening* of A with a structuring element B is expressed as $A \circ B$ and defined by

$$A \circ B = (A \ominus B) \oplus B \quad (4.25)$$

$$= \bigcup_{B+p \subseteq A} B + p. \quad (4.26)$$

□

Figure 4.15 shows simple examples of opening.

Definition 4.4 (closing): The *closing* of A by structuring element B is expressed as $A \bullet B$, and is defined by

$$A \bullet B = (A \oplus B) \ominus B. \quad (4.27)$$

□

Figure 4.16 shows simple examples of closing. As shown in this figure, if B is a disk, it smooths nonconvex corners, and it fills narrow holes.

The principle of duality, in the sense of set operations, holds for opening and closing.

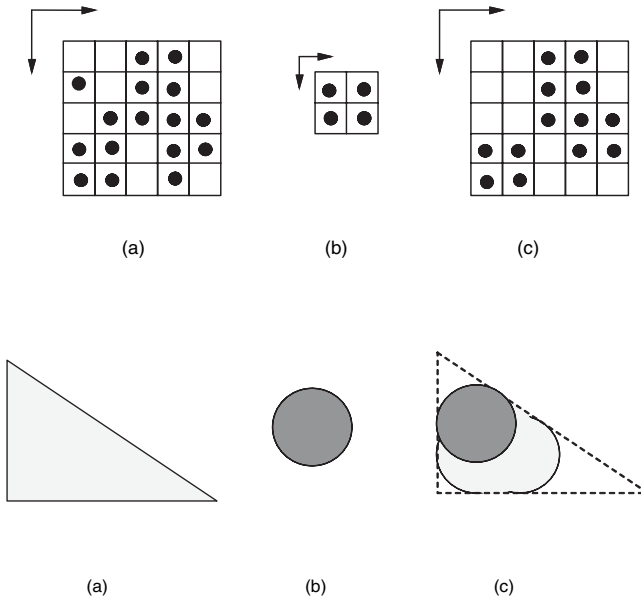


Figure 4.15 Opening in the discrete domain (top row) and in the continuous domain (bottom row): (a) the original pictorial image A ; (b) the structuring element B ; (c) their opening of $S = A \circ B$

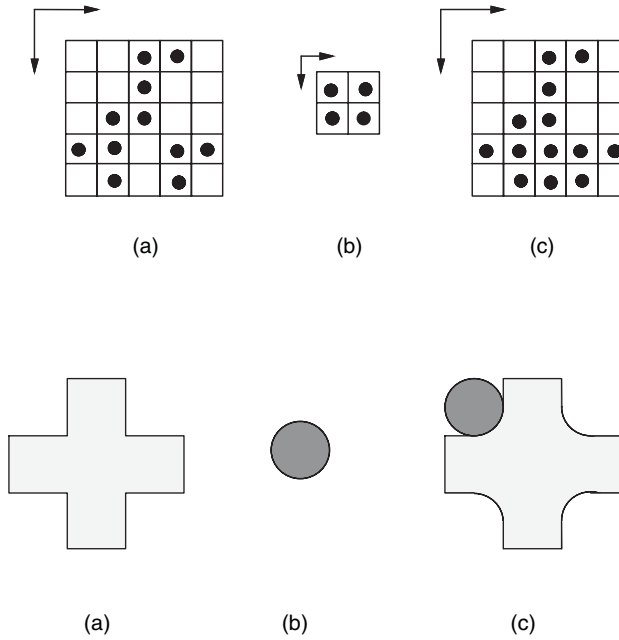


Figure 4.16 Closing in the discrete domain (top row) and in the continuous domain (bottom row): (a) the original pictorial image A ; (b) the structuring element B ; (c) their closing of $S = A \bullet B$

Theorem 4.5 (duality). *Let us consider A and B as sets of \mathbb{G}^2 . Then,*

$$(A \bullet B)^c = A^c \circ \check{B}, \quad (4.28)$$

$$(A \circ B)^c = A^c \bullet \check{B}, \quad (4.29)$$

where A^c is the complement of A , and $\check{B} = \{-b; b \in B\}$.

The proof is based on the fact that

$$(A \oplus B)^c = A^c \ominus \check{B}, \quad (4.30)$$

$$(A \ominus B)^c = A^c \oplus \check{B}. \quad (4.31)$$

Opening and closing have the property of idempotence, which means that more than one operation with the same operands produces the same result as the first one. This can be expressed as follows.

Proposition 4.6.

$$(X \circ B) \circ B = X \circ B, \quad (4.32)$$

$$(X \bullet B) \bullet B = X \bullet B. \quad (4.33)$$

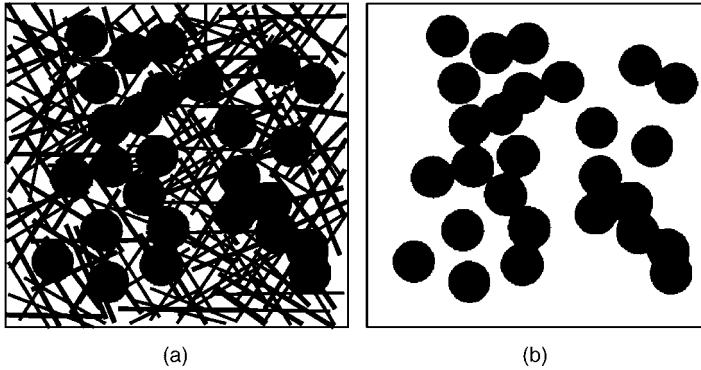


Figure 4.17 (a) A binary image. (b) The opening of (a) with a disk structuring element

As we mentioned at the beginning of this chapter, the erosion \ominus removes all isolated blobs that are smaller than B (see Figure 4.4). However, the size of the remaining blobs is also reduced due to the erosion. Then, to restore the original size, we employ

$$(X \ominus B) \oplus B,$$

which is simply the opening $X \circ B$.

Figure 4.17 shows an application of opening to noise elimination and pattern extraction. Part (a) shows a binary image of disks with an average diameter of 35 pixels in a dense background of short line segments. We can perform an opening to eliminate the line segments. Opening the image of part(a) with a disk structuring element that has a diameter of 13 produces the image shown in part (b) – a surprising, almost perfect, result.

It has been reported that to yield an exact result for $X \circ \text{disk}(r)$, where $\text{disk}(r)$ is a circular disk of radius r , it is better to add more operations to the opening, so that

$$((X \circ \text{disk}(r)) \oplus \text{box}(r_1)) \cap X,$$

where r_1 is slightly smaller than r and $\text{box}(r_1)$ is a square box with side-length r_1 .

For isolated holes or cracks, a dual operation is recommended, of the form

$$((X \bullet \text{disk}(r)) \ominus \text{box}(r_1)) \cap X.$$

4.6.3 Morphology of Multivalued Figures

In morphology, the operations for multivalued or gray-level figures are considered in the following ways. First, the multivalued figure in \mathbb{G}^2 is treated as a binary figure in three-dimensional space \mathbb{G}^3 , in which the additional dimension represents the gray-level of the point coordinated in \mathbb{G}^2 .

A gray-level function $g(\mathbf{x})$ with a coordinate variable \mathbf{x} is given as a set

$$U(g) = \{(\mathbf{x}, t) \in \mathbb{G}^3 ; t \leq g(\mathbf{x})\}. \quad (4.34)$$

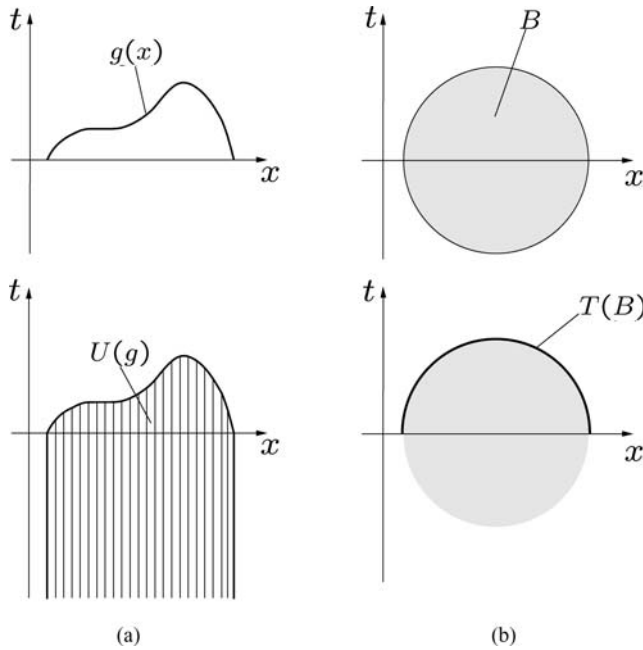


Figure 4.18 The umbra and the top of a multivalued function. (a) The umbra of a function $g(x)$. (b) The top of set B

That is to say, a set of points at \mathbf{x} in \mathbb{G}^3 whose values are smaller than or equal to $g(\mathbf{x})$ is called an *umbra*, and $g(\mathbf{x})$ is defined as a set of the umbras (see Figure 4.18). Here, we treat a multivalued figure as a binary pattern in a space of one higher dimensionality.

Then, we introduce one more definition, that of the *top* $T(B)$ of the umbra $B \in \mathbb{G}^{2+1}$, as

$$T(B) = \sup\{t \in \mathbb{G} ; (\mathbf{x}, t) \in B\}. \quad (4.35)$$

The schematic of the *top* is also shown in Figure 4.18.

From the definition of the top, it is clear that, for a \mathbb{G} -valued function g on \mathbb{G}^2 ,

$$T(U(g)) = g. \quad (4.36)$$

Therefore, morphological operations on multivalued figures are given as a series of operations; that is, a binary morphological operation is first applied to the umbra of the figure, and then the top operation is applied to produce the result. For further details, see the references listed in [37, 65, 66, 67, 89], and so on.

4.6.4 Morphological Expansion

The morphological expansion of a binary shape by a natural number is defined as follows.

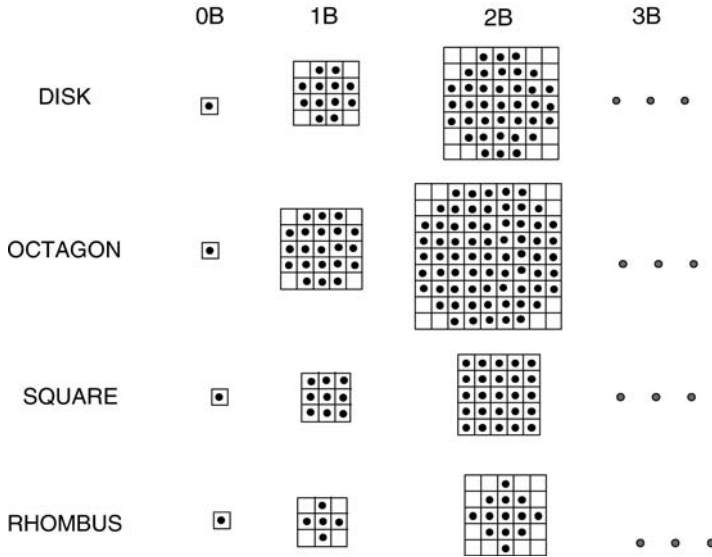


Figure 4.19 Examples of structuring elements B in the discrete domain that are widely used, and their morphological n times expansions nB ($n = 0, 1, 2, \dots$)

Definition 4.5: For a binary shape B , its n times expansion nB is given as

$$nB = \overbrace{B \oplus B \oplus \dots \oplus B}^{n \text{ times}}. \quad (4.37)$$

□

In a regular sense, an r times expansion of a binary continuous shape B is given as

$$rB = \{rb; b \in B\}, \quad (4.38)$$

where $r \geq 0$. When B is convex and $r = n$ (natural number), this becomes just the same as the above morphological expansion. However, for a discrete binary shape, (4.38) produces a sparsely dotted shape over the pixel field, which is hardly interpreted as an expansion of the original shape.

It should be noted that when B is not convex, the similar expansion of nB produces a very different form of a shape. For a nonconvex shape, nB converges to the n times expansion of the convex hull of B .

Here, in Figure 4.19, we show some examples of morphological expansions of binary shapes, which are to be employed as structuring elements in the later experiments.

4.6.5 The Morphological Skeleton and its Properties

The next example of a morphological operation to be introduced here is that of a morphological skeleton. This operation, which is a type of thinning process of a shape along its skeleton,

extracts a skeleton of a binary shape, which expresses the principal structure of the shape. The morphological skeleton has the special property that the original shape can be restored from its skeleton.

In order to simplify this explanation, we will limit our subject to discrete images. For the skeletonization of continuous images, see [66] and [90].

Let X be a discrete binary image of finite size, and let B be a binary pattern, also of finite size, containing the origin $(0, 0)$ within it. Then the skeletonization of a binary image is defined as follows.

Definition 4.6 (the skeletoning of a binary image): Let $N = \max\{n \geq 0; X \ominus nB \neq \emptyset\}$; then S_n , given as the following definition, is called the (morphological) *skeleton element*.

$$S_n = (X \ominus nB) - [(X \ominus nB) \circ B], \quad \text{for } n = 0, 1, \dots, N. \quad (4.39)$$

The union of all S_n – that is,

$$SK(X) = \bigcup_{n=1}^N S_n \quad (4.40)$$

– is called the *morphological skeleton* of X , and a point in $SK(X)$ is called a skeleton point.

The total process of skeletonization is characterized by the sequence of sets

$$ST(X) = (S_0, S_1, S_2, \dots, S_N), \quad (4.41)$$

which is called the *skeleton transform*. □

The above skeleton transform can be interpreted physically as follows. From all the points on the boundary of an original figure, structuring elements start to grow and form progressing wave fronts. Then, the skeleton points arise at the places where two of the wave fronts of this growth meet. That place is equidistant from the starting points of the two waves and becomes an axis point for the local shape. In this sense, the skeleton transform may sometimes be called the *medial axis transform*.

The most important property of the morphological skeleton is that the original figure can be restored from the skeleton.

Theorem 4.7 (reconstruction from the skeleton): Let the skeleton transform of an original shape X be $ST(X) = (S_0, S_1, \dots, S_N)$. Then, it holds that

$$X \circ nB = (((S_N \oplus B) \cup S_{N-1}) \oplus B) \cdots \cup S_n) \oplus nB, \quad 0 \leq n \leq N. \quad (4.42)$$

Since \cup and \oplus are interchangeable, then

$$X \circ nB = \bigcup_{k \leq n} S_k \oplus nB, \quad 0 \leq n \leq N. \quad (4.43)$$

Especially, $n = 0$ gives the original shape itself, as

$$X = ((S_N \oplus B) \cup S_{N-1}) \oplus \cdots \cup S_0, \quad (4.44)$$

or

$$X = \bigcup_{0 \leq n \leq N} S_n \oplus nB. \quad (4.45)$$

(4.42) or (4.43) with $n \neq 0$ gives the opening of X . This is called partial reconstruction of X .

Proof: From

$$\begin{aligned} S_n &= (X \ominus nB) - [(X \ominus nB) \circ B] \\ &= (X \ominus nB) - [(X \ominus (n+1)B) \oplus B] \end{aligned} \quad (4.46)$$

and $X \ominus NB = S_N$, we have

$$\begin{aligned} X \ominus (N-1)B &= [(X \ominus NB) \oplus B] \cup S_{N-1} \\ &= (S_N \oplus B) \cup S_{N-1} \\ X \ominus (N-2)B &= [(X \ominus (N-1)B) \oplus B] \cup S_{N-2} \\ &= \{[(S_N \oplus B) \cup S_{N-1}] \oplus B\} \cup S_{N-2} \\ &\quad \dots \quad \dots \quad \dots \quad \dots \\ X \ominus nB &= ((S_N \oplus B) \cup S_{N-1}) \oplus \dots \cup S_n. \end{aligned}$$

Then, by applying $\oplus nB$ on both sides of the last formula, we obtain (4.42). \square

The reconstruction given by (4.44) can be interpreted physically as follows. From the N th skeleton element to the 0th one, if we start from all of the skeleton points and gradually expand the figure by applying them progressively, from one point to the next, we finally obtain the reconstruction of the original figure.

On the other hand, the reconstruction based on (4.45) first took the largest pattern of nB centered at each skeleton point and inscribed in an original figure, and then summed up those patterns.

Examples of morphological skeletons are shown in Figures 4.20, 4.21, 4.22, and 4.23. Figure 4.20 is a simple binary pattern, and its morphological skeleton is given as Figure 4.21. Another skeleton, of the silhouette of an airplane (Figure 4.22), is shown in Figure 4.23.

4.6.6 Morphological Decomposition of Figures

The decomposition of a figure (or a shape) X means representing it as a sum of partial figures X_1, X_2, \dots , and X_n , under the requirements that:

- the partial figures X_1, X_2, \dots , and X_n clearly have much simpler forms than the original figure X ;
- the decomposition is invariant for translation, scaling, and rotation of the original figure X ; and
- the degree of fineness of the decomposition can be determined arbitrarily.

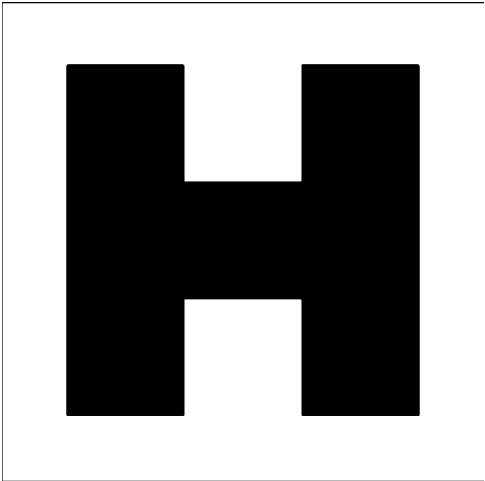


Figure 4.20 A simple binary figure “H” having 128×128 pixels

Here, a simple partial figure is usually of a convex form, typical examples of which are given in Figure 4.19.

Morphological decomposition meeting the above requirements was proposed by Pitas *et al.* [80]. Its algorithm in morphology operations is given as follows.

Definition 4.7 (morphological shape decomposition): Let X be an input figure and B be a structuring element. Then, repeat the next operations until k satisfies $(X - X'_k) \ominus B = \emptyset$:

$$X_i = (X - X'_{i-1}) \circ n_i B \tag{4.47}$$

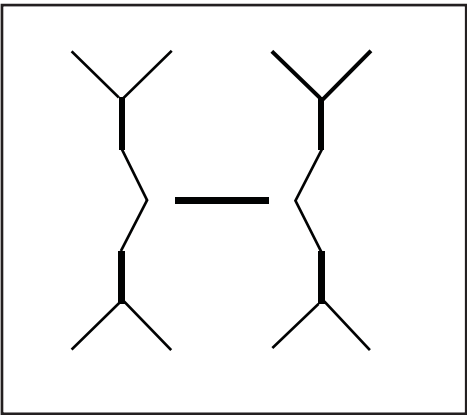


Figure 4.21 The skeleton of Figure 4.20 (“H”) with a structuring element of RHOMBUS

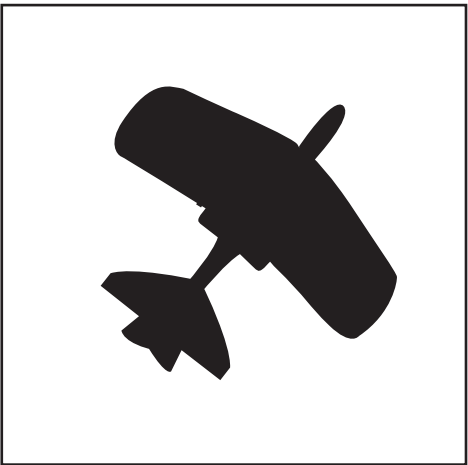


Figure 4.22 An illustration of an “airplane”

and

$$X'_i = \bigcup_{0 < j \leq i} X_j, \tag{4.48}$$

where $X'_0 = \emptyset$, and n_i is the maximum integer for which n_i times expansion $n_i B$ is completely included inside $X - X'_i$. □

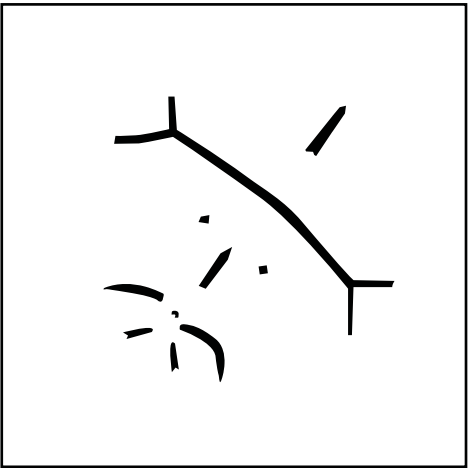


Figure 4.23 The skeleton of Figure 4.22 (an airplane). The structuring element was SQUARE

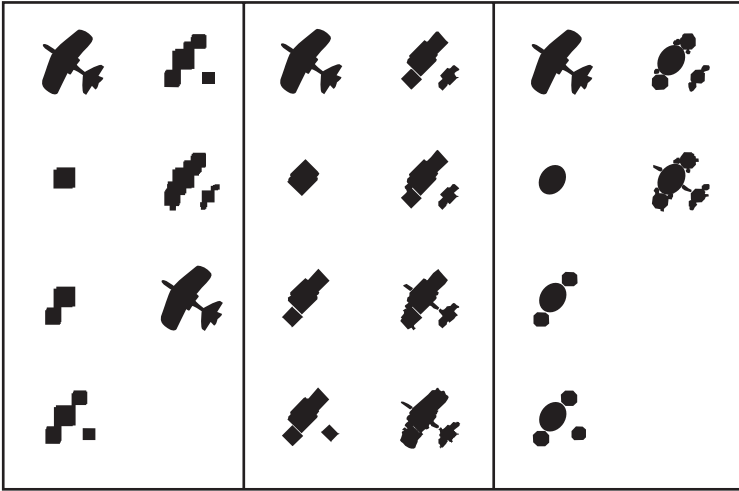


Figure 4.24 The composition of simple decomposition figures $X_1 + \dots + X_N$. From the upper left to the lower right, $N = 1, 2, 3, 4, \dots$, and X_i is the i th decomposition figure using the square, rhombus, and disk, from the left to the right columns, respectively, as the structuring element

Figure 4.24 shows a series of morphological decompositions. The original shape is just the same as Figure 4.22, and is shown in the upper left-hand corner of each column. From the left to the right columns, the structuring elements were the square, rhombus, and disk, respectively.

It is understandable that, from its definition, this decomposition is highly dependent on the structuring element, and that the choice of a better structuring element becomes an important problem – one that Pitas [80] did not discuss.

Figure 4.24 shows that this decomposition proceeds as follows: first, the biggest possible expansion of the structuring element that can be included inside the original figure is searched; then, the same searching process follows for the remaining areas of the original figure, the already occupied area of the searched figures having been subtracted. Therefore, in this process, the series of figures $X_1, X_1 + X_2, X_1 + X_2 + X_3, \dots$ will approach the original figure in greater detail. Then, the interpretation can be put forward that the meaningful decomposition takes place up to some value i , beyond which the process just becomes one of burying “crevices” or filling up small gaps.

Then, let us consider the decompositions of a figure X as

$$X' = X_1 + X_2 + \dots + X_N,$$

at $N = 1, 2, \dots$. If the newly added decomposition X_{N+1} has two or more points that touch the area of X' , this new decomposition only buries the remaining areas. The decompositions that take place before this stage is reached are called *simple decompositions*.

Examples of simple decompositions for the original figure of Figure 4.25 are shown in Figure 4.26, with the disk (circle), octagon, square, and rhombus structuring elements.

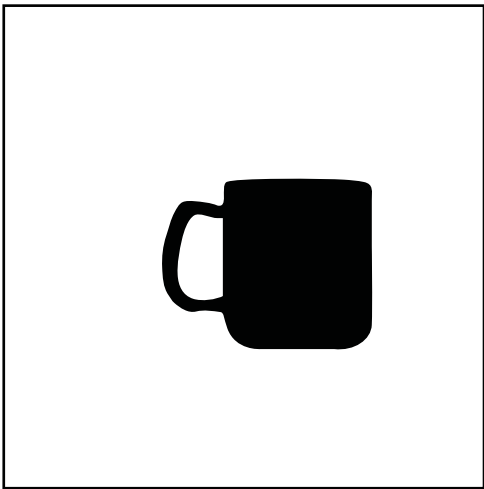
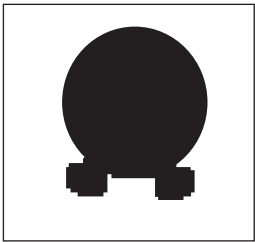
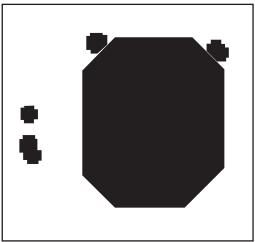


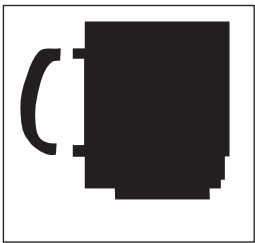
Figure 4.25 The original figure X to be decomposed by various structuring elements



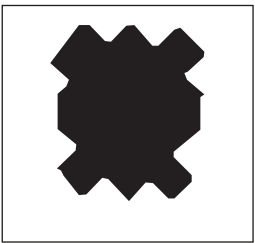
(a) disk



(b) octagon



(c) square



(d) rhombus

Figure 4.26 Simple decompositions of X by the (a) disk, (b) octagon, (c) square, and (d) rhombus structuring elements. The ratios of the areas of the simple decompositions for each structuring element to the original figure are as follows: (a)75.1%, (b)87.6%, (c)91.9%, and (d)76.5%

4.7 The Wealth and Potential of the Minkowski Operators

It is not possible in this introductory chapter, through a shape model of the kind that has been proposed, to expose all the wealth and potential of the Minkowski operators in describing shapes and their motions in space. After worked with the subject for a long time, we personally feel that the significance of the Minkowski operators can be compared to that of the set operators – especially when we are dealing with shapes. Their significance may be better understood if we extend our shape model in the following ways.

4.7.1 Minkowski Operations on Discrete Shapes

Up to this point, we have treated a mixture of “continuous” and “discrete” shape properties. We have considered the discrete property, for example, as a set of discrete points, like dots of ink on a sheet of paper. These “discrete shapes” can nevertheless be thought as objects immersed in continuous spaces. This extension of the shape domain now allows our shape model to describe “tessellated” and “textured” objects (see Figure 4.13). It may also be possible to study the shapes of porous media more systematically [69]. A deeper insight may be gained in other areas too, such as nonoverlapping or partially overlapping decompositions.

4.7.2 Minkowski Operations on Dynamically Varying Shapes

Another implicit assumption in our shape model is that the operands B , T , and S remain invariant throughout. However, some applications demand dynamically varying shapes. For example, graphic artists achieve interesting effects by continuously changing the pressure and angle of the brush as it is moved along a trajectory. A nonrigid solid has an associated time-varying parameter, which should be taken into consideration during its movement. A moving object may rotate at some points in space, to avoid colliding with an obstacle. Time-dependent changes of form are also encountered in fluid dynamics and the growth of organisms.

This kind of problem can be formulated as follows:

$$S = B(t) \oplus T, \quad \text{where } t \in T.$$

This equation captures the fact that the shape of B changes as it is placed at different points of T .

In an earlier paper [25], we briefly touched upon this problem for two-dimensional regions with smooth, closed boundary curves. However, further work is needed to extend our approach to arbitrary shapes and shapes in higher dimensions.

The decomposition problem can be formulated in a similar way:

$$T = S \ominus B(s), \quad \text{where } s \in S,$$

or

$$T = S(b) \ominus B, \quad \text{where } b \in B,$$

and needs to be investigated further.

4.7.3 Inverse Shapes

It has been mentioned that $(\mathcal{P}(E), \oplus)$ forms an “Abelian semigroup.” To make it a group, we need the inverse X^{-1} of every subset X of $\mathcal{P}(E)$, so that

$$X \oplus X^{-1} = o.$$

The introduction of the idea of an inverse shape, although difficult to conceptualize, can give rise to interesting mathematical problems. Will we then be able to decompose a triangle by a circle? We have arrived at a particularly interesting problem, which will be discussed in the following chapters.

We will summarize this chapter by saying that the study of the Minkowski operators in the context of shape description seems to be extremely rewarding, although at present there are still more unknowns than knowns.

5

Arithmetics of Geometric Shape

5.1 The Motivation for a Shape Arithmetic

5.1.1 Does Negative Shape Exist?

Our previous chapter ended up posing a simple question: “Is it possible to do addition and subtraction (or multiplication and division) with geometric shapes as we do in ordinary arithmetic with numbers?” In other words, for a geometric shape, does its inverse – that is, *negative shape* – exist? If this were possible, then we would obtain a remarkable insight into analyzing and synthesizing shapes, just as we have in the case of numbers. Unfortunately, the precise answer to this question is “No,” because the *algebraic structures* of various systems with geometric shapes are, in general, *weaker* than those of systems with numbers. For example, consider some of the most familiar algebraic systems with numbers, such as $(\mathbb{Z}, +)$, $(\mathbb{Q}, +)$, and $(\mathbb{R}, +)$, where \mathbb{Z} , \mathbb{Q} , and \mathbb{R} denote the sets of integers, rational numbers, and real numbers, respectively, and the composition operation “+” denotes the ordinary addition operation. All of these systems are groups. If we exclude the number 0 from all these sets and consider ordinary multiplication “ \cdot ” as the composition operation, the corresponding systems, $(\mathbb{Z} - \{0\}, \cdot)$, $(\mathbb{Q} - \{0\}, \cdot)$, and $(\mathbb{R} - \{0\}, \cdot)$, again turn out to be groups.

On the other hand, take some of the common systems with geometric shapes, such as (\mathcal{G}, \cup) , (\mathcal{G}, \cap) , (\mathcal{G}, \oplus) , or (\mathcal{G}, \ominus) , where \mathcal{G} denotes the set of all geometric shapes (more formally, the set of all subsets of real Euclidean space R^d), and \cup , \cap , \oplus , and \ominus denote, respectively, set union, set intersection, and the Minkowski (vector) addition and decomposition of two sets of points. All of these systems are *monoids* – but not groups. We may recall that a group has a *stronger* structure than a monoid, since a restriction imposed on the elements of a monoid – namely, the existence of an inverse for each element – results in a group.

The following question then naturally arises: “Is it possible to extend our notion of geometric shapes so that with a convenient composition operation we can form an algebraic group-like structure?”

This is indeed a very fundamental question. In this chapter, we shall attempt a much modest task by restricting our domain of shapes to the polygons in E^2 and the polyhedra in E^3 , where E denotes Euclidean space, and approaching the problem in a more intuitive way, rather than becoming strictly mathematical.

To devise a possible approach, we can draw an analogy with the lessons of number systems, as described in the previous chapters. We can start with the system $(\mathbb{N}, +)$, which is a monoid, where \mathbb{N} denotes the set of natural numbers; that is, $\mathbb{N} = \{0, 1, 2, \dots\}$. Let us take a small subset of \mathbb{N} , say $A = \{0, 1, 2\}$, and a composition operation $+$ redefined by the following table:

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

It is evident that the system $(A, +)$ is a group, since each of the elements of A has a unique inverse. You must have realized that we are essentially using *modulo 3* arithmetic. It is well known that the relation “ $a = b \text{ (modulo } m)$ ” is an *equivalence relation* that partitions \mathbb{N} into m nonoverlapping subclasses, called *equivalence classes*. The equivalence classes may allow us to form a group structure, as was done in the above example. Since $(A, +)$ is a group, it is possible to solve an equation such as

$$x + 2 = 1$$

using this small system.

Now, instead of taking a small subset of \mathbb{N} , such as A , if we consider the whole of \mathbb{N} and try to solve an equation of the above form, we are led to the concept of negative integers, $-1, -2, \dots$. The set of natural numbers \mathbb{N} augmented by the set of negative integers forms the set of integers \mathbb{Z} , and the system $(\mathbb{Z}, +)$ is now obtained as a group where both additions and subtractions become possible.

5.1.2 What Form Must Negative Shapes Take?

Now, let us confirm the theories on the extension of the geometric shape concept that we have constructed so far in this book.

We already have the next fundamental theory that tells us about this extension:

Extension of Algebraic Systems: If an algebraic system (A, \circ) where \circ is a binary operation on the element in a set A has a *unit (identity) element* e which satisfies

$$a \circ e = e \circ a = a$$

for any element $a \in A$, and has next three properties that,

1. $a \circ b = b \circ a$, (Commutativity)
2. $(a \circ b) \circ c = a \circ (b \circ c)$, (Associativity)
3. If $a \circ b = a \circ c$, then $b = c$, (Injectivity)

for any elements $a, b, c \in A$, then A can be extended to $A' \supseteq A$ so that all elements $a' \in A'$ have their own respective inverses in A' ; that is, $\exists b' \in A'$, for $\forall a' \in A$ and $a' \circ b' = e$.

Bourbaki, *Algebra I* [11]; see also Section 3.2

Let us recall our system of (\mathcal{G}, \oplus) . A single point, say $\{o\}$ (the origin), is a unit element of this system, because, for any geometric shape S , $S \oplus \{o\} = \{o\} \oplus S = S$. (It should be noted that the origin $\{o\}$ is not a unique unit element for the operation of \oplus . Actually, every single point $\{p\}$ can be a unit element. This is another substantial weak point in our theory.)

Now, we denote the set of all the convex polygons by \mathcal{C} ; that is, $\mathcal{C} \subset \mathcal{G}$. Then, all of the above properties 1, 2, and 3 hold in the system (\mathcal{C}, \oplus) , as was shown in the previous chapter. This means that the set of all convex polygonal shapes \mathcal{C} can be extended to include the inverses of the shapes. For example, for a rectangular shape R , the above fundamental theory says that we could consider its inverse shape R^{-1} so that $R \oplus R^{-1} = \{o\}$ (or a single point). Also, a circular shape C could have its inverse shape C^{-1} .

So, the first problem here is: What, in reality, are the inverse shapes R^{-1} and C^{-1} ? The next question is: “How is the inverse operation actually defined to generate R^{-1} and C^{-1} from R and C , respectively?” Especially, how does the inverse operation of \oplus relate to \ominus ? And the third problem is: “If we have a proper extension of the set of the convex polygons \mathcal{C} , which includes their negative shapes, can it be further extended for all the geometric shapes in \mathcal{G} ?”

Just as we said at the beginning of this section, the general answer to the third question is “No.” Therefore, the question must be rewritten as follows: “To what extent can the set of the convex polygons \mathcal{C} be defined such that each of the element shapes has its own inverse?”

We present solutions to some of these problems in this chapter, where negative shapes in terms of Minkowski addition are defined as being closely related to Minkowski decomposition. We will only treat convex objects in this chapter: the results will be extended to nonconvex objects in the next chapter.

5.2 Morphology and the Theory of Numbers

5.2.1 Morphology for High-Level Vision

The two operations of Minkowski addition \oplus (also called dilation) and Minkowski decomposition \ominus (also known as erosion), introduced in the previous chapter, form the kernel of all other morphological operations.

To this day, morphological operations are mostly used for low-level image processing; that is, for early processing of binary or gray-scale discrete images [90]. Their applications in representing or understanding two- or three-dimensional geometric objects for the purpose of high-level object recognition are very limited. This is certainly an enigmatic situation, considering the fact that these operations are essentially functions of the form $f : \mathcal{G}(E^d) \rightarrow \mathcal{G}(E^d)$, where $\mathcal{G}(E^d)$ denotes the power set of real d -dimensional Euclidean space E^d ; their definitions do not discriminate between whether the underlying sets are discrete images or continuous geometric objects. Moreover, a number of researchers have pointed out the relevance of these operations in high-level vision applications such as geometric modeling, spatial planning, description and understanding of biological forms, crystallography and textured object modeling, and so on [27, 28, 29, 60, 84]. The close resemblance between the representation of a generalized cylinder and the Minkowski addition operation has also been observed.

One factor that may account for this situation has to do with the computational problems associated with Minkowski operations on continuous objects. Note that any binary image can be modeled as a set consisting of a finite number of discrete points. Therefore, in computing the Minkowski sum $A \oplus B$ of two binary images A and B , we can directly use the set operations of Minkowski addition:

$$A \oplus B = B \oplus A = \bigcup_{b \in B} A_b = \bigcup_{a \in A} B_a.$$

Similarly, Minkowski decomposition $A \ominus B$ of a binary image A can be computed by means of

$$A \ominus B = \bigcap_{-b \in \tilde{B}} A_{-b}.$$

On the other hand, a continuous “well-formed” object (such as a polygon, circle, ellipse, etc. in two dimensions, or a polyhedron, sphere, ellipsoid, etc. in three dimensions) cannot be specified as a collection of a finite number of points. In general, such an object, say A , is specified by its oriented boundary, say ∂A . This necessitates computation of the boundary $\partial(A \oplus B)$ or $\partial(A \ominus B)$ of the products from the boundaries ∂A and ∂B of the operands. But it can immediately be seen that $\partial(A \oplus B) \neq \partial A \oplus \partial B$ and also $\partial(A \ominus B) \neq \partial A \ominus \partial B$.

In general, $\partial(A \oplus B) = f(\partial A, \partial B)$, and also $\partial(A \ominus B) = g(\partial A, \partial B)$, where “ f ” and “ g ” denote some complicated functions whose computations turn out to be quite nontrivial.

5.2.2 The Resemblance between Morphology and the Theory of Numbers

We can observe a remarkable similarity between some number-theoretic results and morphological theorems, particularly in the domain of convex objects. Let us consider the number system $(\mathbb{N}, \cdot, /)$, where \mathbb{N} denotes the set of natural numbers $\{0, 1, 2, \dots\}$ and “ \cdot ” and “ $/$ ” denote the multiplication and division operations, respectively. Since, within \mathbb{N} , the division operation is not an exact inverse, but only a “restricted” inverse of the multiplication operation, we can define the division of a number m by another number n as $\lfloor m/n \rfloor$; the floor function notation $\lfloor x \rfloor$ means the greatest integer less than or equal to x . This number system can be compared with the geometric system $(\mathcal{K}, \oplus, \ominus)$, where \mathcal{K} denotes the set of all compact convex subsets of E^d . We refer to Table 5.1, where $A, B, C \in \mathcal{K}$ and $m, n, p \in \mathbb{N}$.

From the table, it appears as though $A \oplus B$ in the convex domain is similar to $m \cdot n$, whereas $A \ominus B$ is similar to $\lfloor m/n \rfloor$. In this chapter, we do not attempt to seek why such a resemblance exists. Instead, we adopt the computational guideline that can be derived from this resemblance.

Table 5.1 The resemblance between the morphological system and the integer number system

	System $(\mathcal{K}, \oplus, \ominus)$	System $(\mathbb{N}, \cdot, /)$
1.	$A \oplus B = B \oplus A$	$m \cdot n = n \cdot m$
2.	$A \oplus (B \oplus C) = (A \oplus B) \oplus C$	$m \cdot (n \cdot p) = (m \cdot n) \cdot p$
3.	$A \supseteq B$ implies $C \ominus A \subseteq C \ominus B$	$m > n$ implies $\lfloor p/m \rfloor \leq \lfloor p/n \rfloor$
4.	$A \subseteq B \ominus C$ iff $B \supseteq A \oplus C$	$m \leq \lfloor n/p \rfloor$ iff $n \geq m \cdot p$
5.	$A \oplus B = ((A \oplus B) \ominus B) \oplus B$	$m \cdot n = \lfloor (m \cdot n)/n \rfloor \cdot n$
6.	$A \ominus B = ((A \ominus B) \oplus B) \ominus B$	$\lfloor m/n \rfloor = \lfloor (\lfloor m/n \rfloor \cdot n)/n \rfloor$
7.	$(A \oplus B) \ominus C = A \ominus (B \oplus C)$	$\lfloor \lfloor m/n \rfloor / p \rfloor = \lfloor (m/(n \cdot p)) \rfloor$
8.	$A \oplus (B \ominus C) \subseteq (A \oplus B) \ominus C$	$m \cdot \lfloor n/p \rfloor \leq \lfloor (m \cdot n)/p \rfloor$
9.	$(A \ominus B) \oplus C \subseteq (A \oplus C) \ominus B$	$\lfloor m/n \rfloor \cdot p \leq \lfloor (m \cdot p)/n \rfloor$
10.	$(A \ominus B) \oplus B \subseteq A$	$\lfloor m/n \rfloor \cdot n \leq m$
11.	$(A \ominus B) \oplus B$ $= (((A \ominus B) \oplus B) \ominus B) \oplus B$	$\lfloor m/n \rfloor \cdot n$ $= \lfloor (\lfloor m/n \rfloor \cdot n)/n \rfloor \cdot n$
12.	If $A \oplus B = A \oplus C$, then $B = C$	If $m \cdot n = m \cdot p$, then $n = p$

The value of m/n is, in general, a real number that consists of the “integer part” $\lfloor m/n \rfloor$ and the “fractional part” $m/n - \lfloor m/n \rfloor$. In the integer number domain, at the time of division we discard that fractional part and take only the integer part. We find that the morphological operations can also be treated in a similar way. The idea is to devise an operation that is the “exact” inverse of the Minkowski addition operation, such that the application of this operation on two convex objects will produce a generalized geometric object (like producing a rational number by the division of two integers) that has a physically “realizable part” (analogous to the integer part) and a “nonrealizable part” (analogous to the fractional part). The Minkowski decomposition $A \ominus B$ can then be thought of as the realizable part of the object after the nonrealizable part is discarded.

The nonrealizable part of a generalized object will be referred to as the *negative* part. (We will avoid using terms such as “nonrealizable” or “fractional” part, since they have some widely accepted literal meanings.) The realizable part – that is, any ordinary geometric object – can be called *positive*.

In our subsequent discussions, we will show how such a computational strategy can be devised.

5.3 Boundary Representation by Support Functions for Morphological Operations

5.3.1 The Support Function Representation

Our starting point in tackling this problem is a basic result given by Shephard [93], which is concerned with the boundary addition of d -dimensional convex polytopes. Informally, a convex polytope in E^d is the d -dimensional analogue of a convex polygon in E^2 or a convex polyhedron in E^3 . Mathematically, a polytope is the convex hull of a finite set of points in E^d . Shephard’s result is stated in terms of the faces of polytopes. In this subsection, we briefly discuss the formal notion of the faces of a d -dimensional polytope, and then we state the main result in the next subsection.

A face of a convex polytope A in E^d is defined in terms of the *supporting function* and the *supporting hyperplane* of the polytope. The representation of a convex body by means of its support function is a classical approach [8,34] that was introduced by Minkowski in 1903, and has been widely used by mathematicians since that time.

Let $A \in E^d$ be a convex body (i.e., a nonempty compact convex set) in real Euclidean d -dimensional space. The supporting function $H(A, v)$ of A for all $v \in E^d$ (provided that $v \neq 0$ – that is, v – is an arbitrary vector different from the origin O) is defined by

$$H(A, v) = \sup\{\langle a, v \rangle \mid a \in A\}, \quad (5.1)$$

where $\langle a, v \rangle$ denotes the scalar (or inner) product of the vectors a and v , and “sup” denotes the “supremum” or “least upper bound.”

Since $H(A, \lambda v) = \lambda H(A, v)$ for any real number $\lambda > 0$, the support function $H(A, v)$ is completely determined by its value on the unit sphere $\|v\| = 1$. Thus, if u denotes a unit vector – that is, $u \in S^{d-1}$, where S^{d-1} is the unit sphere in E^d with its center at the origin – it is most convenient to use the function $H(A, u)$ as the support function of A . $H(A, u)$ is a *complete representation* of the convex body A , since the values of $H(A, u)$ for all $u \in S^{d-1}$ completely

specify A in the following way:

$$A = \{x \in E^d \mid \langle x, u \rangle \leq H(A, u) \text{ for all } u \in S^{d-1}\}, \quad (5.2)$$

which, in words, means that A is the intersection of all the half-spaces $\langle x, u \rangle \leq H(A, u)$.

Although a large part of the theory of convex bodies in mathematics uses $H(A, u)$ as the standard representation of a convex body A , in the fields of computer vision, graphics, and image processing, the support function representation is less frequently used. One primary reason is that the function $H(A, u)$ is, in general, a continuous function of u , whose closed-form specification may not be readily available. It may also appear that the $H(A, u)$ -representation is less intuitive than the boundary representation or half-space representation of convex bodies.

5.3.2 The Support Function is a Signed Distance

If, for some $u \neq 0$, we have $H(A, u) < \infty$ (this condition ensures that A is bounded), then the hyperplane

$$L(A, u) = \{p \in E^d \mid \langle p, u \rangle = H(A, u)\} \quad (5.3)$$

is called the *supporting hyperplane* of A with *outer/outerward normal* u . In E^2 , the supporting hyperplane becomes the supporting line of a convex polygon A with outer normal u , while in E^3 it is the supporting plane of a convex polyhedron A .

It is easy to see in Figure 5.1 that the support function $H(A, u)$ is precisely the “signed distance” from the origin O to the supporting hyperplane $L(A, u)$. This distance is to be considered positive if A and the origin lie on the same side of the supporting hyperplane, negative if A and the origin are separated by the supporting hyperplane, and zero if the origin lies in the supporting hyperplane. It is, therefore, convenient for all practical purposes to assume that the origin lies in the interior of A , so that the function $H(A, u)$ is positive for every u .

To provide examples, we consider three simple two-dimensional convex figures – a unit circle having its center at the origin, a triangle, and an ellipse – and show their corresponding $H(A, u)$ ’s in Figure 5.2. Note that for the unit circle whose center is at O , $H(A, u) = 1$ for all u .

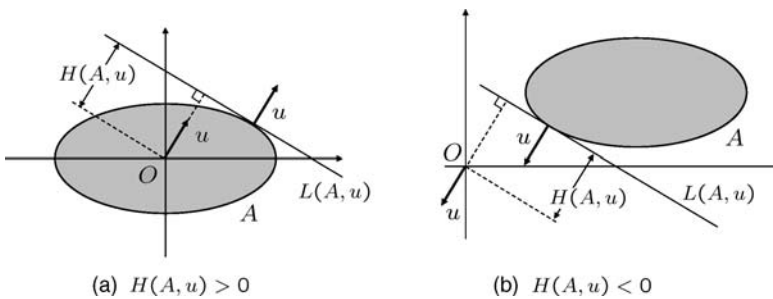


Figure 5.1 The support function $H(A, u)$ is the signed distance from the origin O to the hyperplane $L(A, u)$. It is (a) positive ($H(A, u) > 0$) if A and the origin lie on the same side of the supporting hyperplane, and (b) negative ($H(A, u) < 0$) if A and the origin are separated by the supporting hyperplane

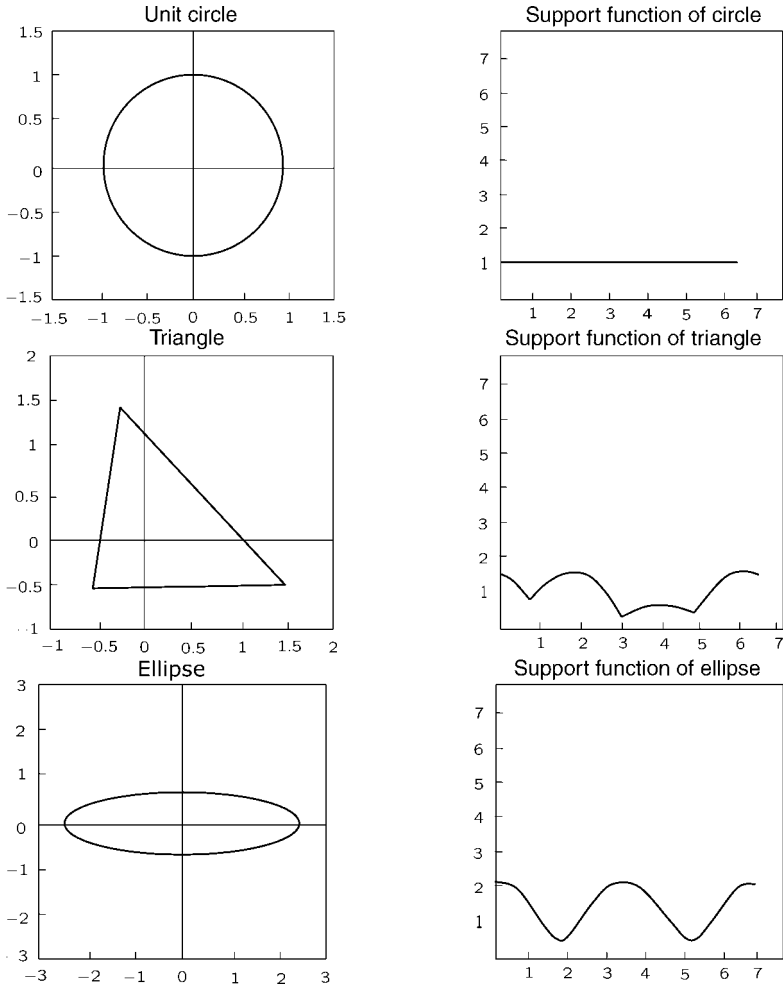


Figure 5.2 The support function representations of some typical convex figures; since in E^2 a unit vector $u = (\cos \theta, \sin \theta)$, it is specified in the graph by the angle θ (in radians) along the x -axis and the corresponding value of $H(A, u)$ along the y -axis

In fact, for some simple convex bodies, we can obtain closed-form representations of the support functions as follows:

1. For a singleton point set $\{a\}$ in E^d , $H(a, v) = \langle a, v \rangle$.
2. For a ball B_α having radius α and center O , $H(B_\alpha, v) = \langle \alpha \frac{v}{\|v\|}, v \rangle = \alpha \|v\|$.
3. For a line segment L_{ab} joining points a and b ,

$$H(L_{ab}, v) = \max(\langle a, v \rangle, \langle b, v \rangle).$$

5.3.3 From Support Function Representation to Boundary Representation and Vice Versa

Assume that the support function $H(A, u)$ of a convex body A is given for all $u \in S^{d-1}$. How do we determine the boundary points of A ? We answer this question by following the approach given in [8]. The boundary points of A , where the outer normal is either u or parallel to u , are precisely the set of points

$$F(A, u) = L(A, u) \cap A. \quad (5.4)$$

$F(A, u)$ is termed the face of A having outer normal u . Obviously, the dimension of $F(A, u)$ is at most $d - 1$.

Figure 5.3 shows examples of faces in E^2 and E^3 with their supporting functions and supporting hyperplanes.

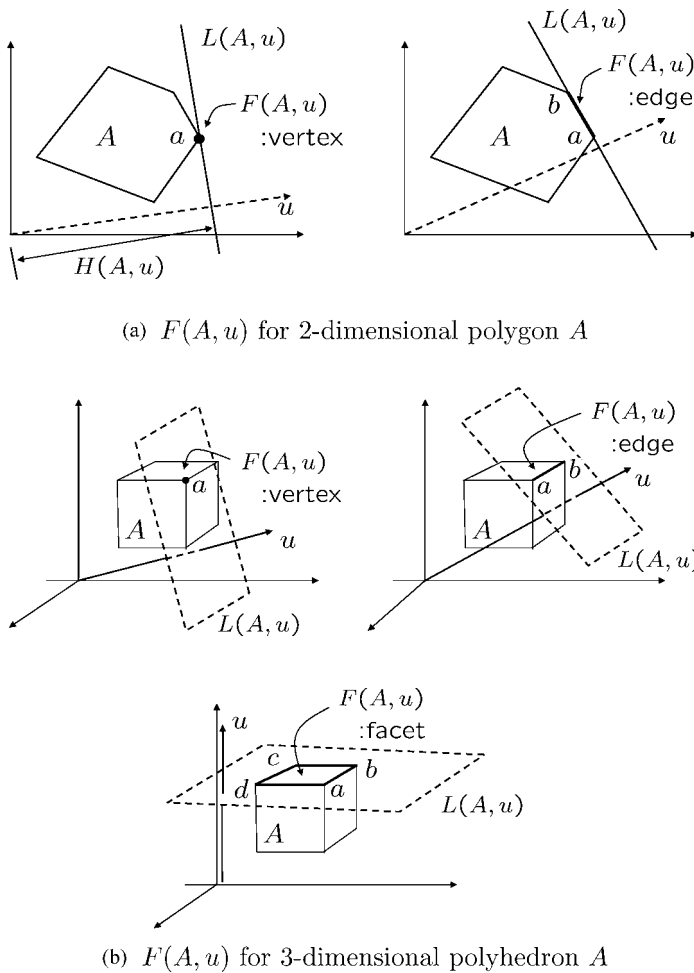


Figure 5.3 The faces $F(A, u)$ of (a) a two-dimensional polygon and (b) a three-dimensional polyhedron

Now let u_k be some fixed direction, and let $F(A, u_k)$ be the face of A that has the outer normal u_k . Our task is to determine $F(A, u_k)$. Since $F(A, u_k)$ is a subset of the supporting hyperplane $L(A, u_k)$, for any point $x \in F(A, u_k)$ it must satisfy (because of (5.3))

$$\langle x, u_k \rangle = H(A, u_k). \quad (5.5)$$

Moreover, $F(A, u_k)$ is also a subset of the convex body A . So a point x of $F(A, u_k)$ must also satisfy, for all u , all of the following inequalities (see (5.2)):

$$\langle x, u \rangle \leq H(A, u). \quad (5.6)$$

Now, if w is a unit vector in an arbitrary direction and $u = u_k + \lambda w$, where $\lambda > 0$, then using (5.5) and (5.6) we can write

$$\langle x, w \rangle \leq \frac{H(A, u_k + \lambda w) - H(A, u_k)}{\lambda}. \quad (5.7)$$

Therefore, letting $\lambda \rightarrow 0$,

$$\langle x, w \rangle \leq H'_w(A, u_k), \quad (5.8)$$

where $H'_w(A, u_k)$ is the directional derivative of $H(A, u)$ at $u = u_k$, in the direction of the unit vector w .

If we consider all of the w 's (i.e., unit vectors in all directions), then the inequalities of (5.8) represent the intersection of the corresponding half-spaces, and that intersection is precisely the face $F(A, u_k)$.

Thus we arrive at the following proposition.

Proposition 5.1. *If $H(A, u)$ is the support function of a convex body A , then the face $F(A, u_k)$ of A having the outer normal u_k has the support function $H'_{u_k}(A, u_k)$.*

5.3.4 Necessary and Sufficient Conditions for a Function to be a Support Function

The support function $H(A, v)$ is a scalar function of the vector v and hence, in the present case, a mapping from E^d to R . A natural question is: "Which functions from E^d to R could be characterized as support functions?" This question is particularly important to us since, from the next section onward, we will be concerned as to whether or not a function $H(A, v) * H(B, v)$ resulting from some operation $*$ on the given support functions $H(A, v)$ and $H(B, v)$ is a valid support function. Consider, for example, the situation shown in Figure 5.4, where no object could have all of the support function values $H(A, u_1) * H(B, u_1)$, $H(A, u_2) * H(B, u_2)$, $H(A, u_3) * H(B, u_3)$, $H(A, u_4) * H(B, u_4)$, since the supporting line corresponding to the value $H(A, u_3) * H(B, u_3)$ is a redundant line.

For characterization of the support function, we can state the following result.

Proposition 5.2. (a) *Every real-valued function $F(v)$ defined for all $v \in E^d$ and satisfying the properties*

- (i) $F(o) = 0$,
- (ii) $F(\lambda v) = \lambda F(v)$, for $\lambda > 0$, and

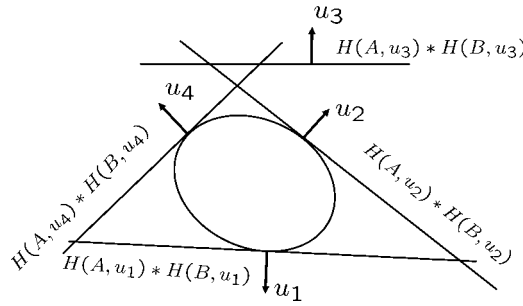


Figure 5.4 An example where the resulting function $H(A, v) * H(B, v)$ cannot be a valid support function; the supporting line corresponding to the value $H(A, u_3) * H(B, u_3)$ is a redundant line

- (iii) $F(v + w) \leq F(v) + F(w)$, is a support function of a convex body.
 (b) If $F(v)$ is a support function, then the convex body that it represents must be the intersection of all the half-spaces $\langle x, v \rangle \leq F(v)$.

Part (a) is a classical result, the proof of which can be found in [8]. Part (b) follows easily from (5.2). For a review on the topic of consistency checking of the support function, see also [48]. Condition (iii) in the proposition – that is, $F(v + w) = F(v) + F(w)$ – appears to be the most nonintuitive compared to the other two conditions. However, its connection with the question of redundant half-spaces can be easily demonstrated. You are recommended to visualize this condition for yourself.

5.4 Geometric Operations by Means of Support Functions

We assume that the support functions $H(A, u)$ and $H(B, u)$ of two convex bodies A and B are given. We will now show that some simple algebraic operations on the support functions result in fairly complicated geometric operations involving the bodies A and B .

5.4.1 MAX and MIN Operations (Convex Hull and Intersection)

5.4.1.1 The MAX Operation

The MAX operation is defined as

$$\max\{H(Au), H(B, u)\},$$

for every $u \in S^{d-1}$, where “ $\max\{\alpha, \beta\}$ ” specifies the maximum of the two real numbers α and β .

It is not difficult to prove that the MAX operation results in the *convex hull* operation of the union of A and B . Toward that end, we need the following proposition.

Proposition 5.3. *If $H(A, u)$ and $H(B, u)$ are the support functions of two convex bodies A and B , then the inequality*

$$H(A, u) \leq H(B, u), \quad \text{for all } u,$$

holds if and only if $A \subseteq B$.

The proof of the proposition follows immediately from (5.1) and (5.2) (see also [8]).

We can now state and prove the main result concerning the MAX operation.

Proposition 5.4. (a) *The function $\max\{H(A, u), H(B, u)\}$ is a support function.*

(b) *$\max\{H(A, u), H(B, u)\} = H(C, u)$, where $C = \text{conv}(A \cup B)$. (Here, $\text{conv}(X)$ denotes the convex hull of the set X .)*

Proof: (a) See Proposition 5.2. Conditions (i) and (ii) obviously hold for $\max\{H(A, u), H(B, u)\}$, since they hold for both $H(A, u)$ and $H(B, u)$. Only condition (iii) – that is, the subadditivity condition – needs to be proved.

Let us write $F(u) = \max\{H(A, u), H(B, u)\}$, and assume that u_1, u_2 are two arbitrary unit vectors. Then,

$$\begin{aligned} F(u_1) + F(u_2) &= \max\{H(A, u_1), H(B, u_1)\} + \max\{H(A, u_2), H(B, u_2)\} \\ &\geq H(A, u_1) + H(A, u_2) \\ &\geq H(A, u_1 + u_2), \end{aligned}$$

since $H(A, u)$ is a support function.

Similarly, $F(u_1) + F(u_2) \geq H(B, u_1 + u_2)$.

Furthermore, since $F(u_1 + u_2) = \max\{H(A, u_1 + u_2), H(B, u_1 + u_2)\}$, $F(u_1 + u_2)$ is either equal to $H(A, u_1 + u_2)$ or equal to $H(B, u_1 + u_2)$.

Therefore, $F(u_1) + F(u_2) \geq F(u_1 + u_2)$, and hence $\max\{H(A, u), H(B, u)\}$ is a support function of some convex body – say, C .

(b) Let $\max\{H(A, u), H(B, u)\} = H(C, u)$, where C is some convex body. Since $H(C, u) \geq H(A, u)$ and also $H(C, u) \geq H(B, u)$ for all u , according to Proposition 5.3, $C \supseteq A \cup B$.

Let us assume that the convex hull $\text{conv}(A \cup B)$ is not C , but that $\text{conv}(A \cup B) = C'$, and that C' is strictly smaller than C ; that is, $C' \subset C$. Therefore, according to Proposition 5.3, $H(C', u) \leq H(C, u)$, for all u . Let us, for some u , say that u_1 , $H(C', u)$ is strictly smaller than $H(C, u)$; that is, $H(C', u_1) < \max\{H(A, u_1), H(B, u_1)\}$. But that is not possible, which means that $C = C'$. \square

We show two convex polygons A and B (in the xy -space) in Figure 5.5(a) and their support function representations (i.e., A and B in the θ_ρ -space) in Figure 5.5(b). The MAX operation of the support functions and the resulting polygon, which is equal to $\text{conv}(A \cup B)$ in the θ_ρ -space, are shown in Figure 5.5(c) (heavy curves). The supporting lines corresponding to each respective curve are shown in Figure 5.5(d), as heavy lines, and then the resulting MAX convex polygon of A and B is given in Figure 5.5(e).

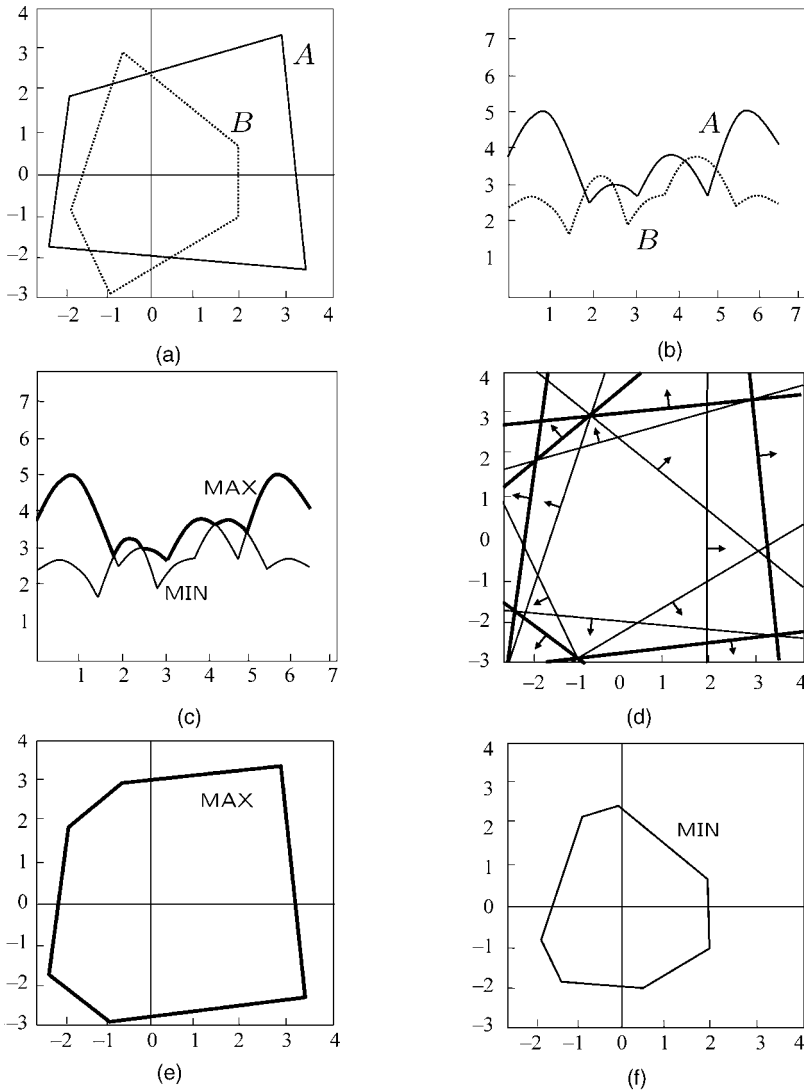


Figure 5.5 A demonstration of the MAX and MIN operations: (a) the input convex polygons A and B (in the xy -space); (b) their support function representations (the θ_ρ -space); (c) $\max\{H(A, u), H(B, u)\}$ (heavy curves) and $\min\{H(A, u), H(B, u)\}$ (light curves) (the θ_ρ -space); (d) the supporting lines corresponding to $\max\{H(A, u), H(B, u)\}$ (heavy lines), and $\min\{H(A, u), H(B, u)\}$ (light lines); (e) the resulting MAX convex polygon (in the xy -space) represented by $\max\{H(A, u), H(B, u)\}$; and (f) the MIN convex polygon (in the xy -space) represented by $\min\{H(A, u), H(B, u)\}$

5.4.1.2 The MIN Operation

The MIN operation is similarly defined as

$$\min\{H(A, u), H(B, u)\},$$

for every $u \in S^{d-1}$, where “ $\min(\alpha, \beta)$ ” denotes the minimum of the two real numbers α and β .

It is not difficult to show that the MIN operation performs the intersection operation $A \cap B$. But unlike the previous case, $\min\{H(A, u), H(B, u)\}$ is not a support function. As a result, some of the supporting hyperplanes defined by $\min\{H(A, u), H(B, u)\}$ may be redundant. But the common intersection of all the half-spaces (some of which may be redundant, corresponding to the redundant hyperplanes) defined by the function $\min\{H(A, u), H(B, u)\}$ will result in $A \cap B$.

To give an example, we consider the same two polygons as presented in Figure 5.5(a), and show the function $\min\{H(A, u), H(B, u)\}$ in Figure 5.5(c) (light curves). The corresponding supporting lines are depicted in Figure 5.5(d) (light lines). Note that some of the supporting lines are redundant. However, the common intersection of the half-spaces defined by all the supporting lines is $A \cap B$, which is shown in Figure 5.5(f).

5.4.2 Morphological Operations in Boundary Representation

The following result concerning Minkowski addition of convex polytopes can now be derived by using the boundary representation formalized in the previous subsection.

Theorem 5.5. *If A and B are two convex polytopes in E^d , then, for every $u \in E^d$,*

$$H(A \oplus B, u) = H(A, u) + H(B, u) \quad (5.9)$$

and

$$F(A \oplus B, u) = F(A, u) \oplus F(B, u). \quad (5.10)$$

Equation (5.9) comes directly from the definition of Minkowski addition \oplus ; $A \oplus B = \{a + b \mid a \in A, b \in B\}$. Then, substituting (5.9) into (5.3) and (5.4), we have (5.10). A detailed proof is given either by Grünbaum [34] or Kelly and Weiss [50], but it is not difficult, and we leave it to the readers.

Let us denote the boundary of S by ∂S . Then, because ∂S consists of all the faces of $F(S, u)$,

$$\partial S = \bigcup_{u \in S^{d-1}} F(S, u).$$

This implies that, from (5.10),

$$\partial(A \oplus B) = \bigcup_{u \in S^{d-1}} (F(A, u) \oplus F(B, u)). \quad (5.11)$$

You may wonder why the \oplus operation appears both on the left- and right-hand sides of (5.11). We shall, however, argue that the computation of the right-hand side finally reduces to two simple operations on real numbers: sorting of real numbers, and addition of real numbers. We will show this first for polygons – that is, in E^2 – and then for polyhedra; that is, E^3 .

5.5 Morphological Operations on Convex Polygons

5.5.1 Computation by Means of Support Function Vectors

The boundary of a convex polygon A in E^d can be precisely defined by means of the supporting function of the polygon. From Theorem 5.5, we can immediately infer the following. For the addition of convex polygons, it is not necessary to compute the supporting function $H(A \oplus B, u)$ for every $u \in S^1$ (S^1 in E^2 is simply a unit circle). Since a convex polygon is completely specified by its oriented edges, it is sufficient to consider only those u 's whose directions are the same as the outer normal directions of the edges of the summand polygons.

Let us, therefore, consider the class of convex polygons whose edges have the same outer normal vectors. In other words, any two polygons belong to this class have pairwise parallel and similarly directed edges. Note that every convex polygon in E^2 can be included in this class by means of introducing edges of zero length. Let $\mathcal{C}(U)$ denote this class, where U is the ordered set of outer normal vectors of the edges; that is, $U = \{u_1, \dots, u_n\}$. In other words, if any convex polygon $A \in \mathcal{C}(U)$, then

$$A = \left\{ p \in E^2 \mid \langle p, u_i \rangle \leq \eta_i \ (i = 1, \dots, n) \right\} \quad (5.12)$$

for some $\eta_i \in R$ ($i = 1, \dots, n$), where R denotes the set of real numbers. Note that η_i is simply the value of the supporting function of A in the direction of the outer normal u_i .

As we know, each η_i along with u_i specifies a closed half-space (in two dimensions, a "half-space" is generally called a "half-plane"), and the intersections of all these half-spaces for $i = 1, \dots, n$, in turn, specify all of the edges of A . Therefore, once U is given, A is specified completely by the vector (η_1, \dots, η_n) . We call this the *supporting function vector* of A , and denote it by $h(A)$.

From Theorem 5.5, we can now easily derive the following result concerning convex polygons.

Proposition 5.6. *If two convex polygons $A, B \in \mathcal{C}(U)$ are represented by the supporting function vectors $(\eta_1^A, \dots, \eta_n^A)$ and $(\eta_1^B, \dots, \eta_n^B)$, respectively, then their Minkowski sum $A \oplus B$ is specified completely by the vector $(\eta_1^A + \eta_1^B, \dots, \eta_n^A + \eta_n^B)$.*

Proposition 5.6 implies that Minkowski addition of two convex polygons can be seen as the vector addition $h(A) + h(B)$ of two points $h(A)$ and $h(B)$ in an n -dimensional space.

Let us clarify the idea described so far by means of an example (Figure 5.6). As is evident from the example, Minkowski addition of two convex polygons is quite straightforward by means of the supporting function vector $h(A) + h(B)$.

From this observation, we can now define the "exact" inverse of Minkowski addition. In computing $A \ominus B$, we shall first compute the vector $h(A) + (-h(B))$. This vector, like the vector $h(A) + h(B)$, again specifies n half-spaces having outer normals u_i 's, but some of the half-spaces may become redundant in this case. It then turns out that the Minkowski decomposition $A \ominus B$ discards those redundant half-spaces and considers the rest. Let us state this result as the following proposition.

Proposition 5.7. *If two convex polygons $A, B \in \mathcal{C}(U)$ are represented by the supporting function vectors $(\eta_1^A, \dots, \eta_n^A)$ and $(\eta_1^B, \dots, \eta_n^B)$, respectively, then their Minkowski decomposition*

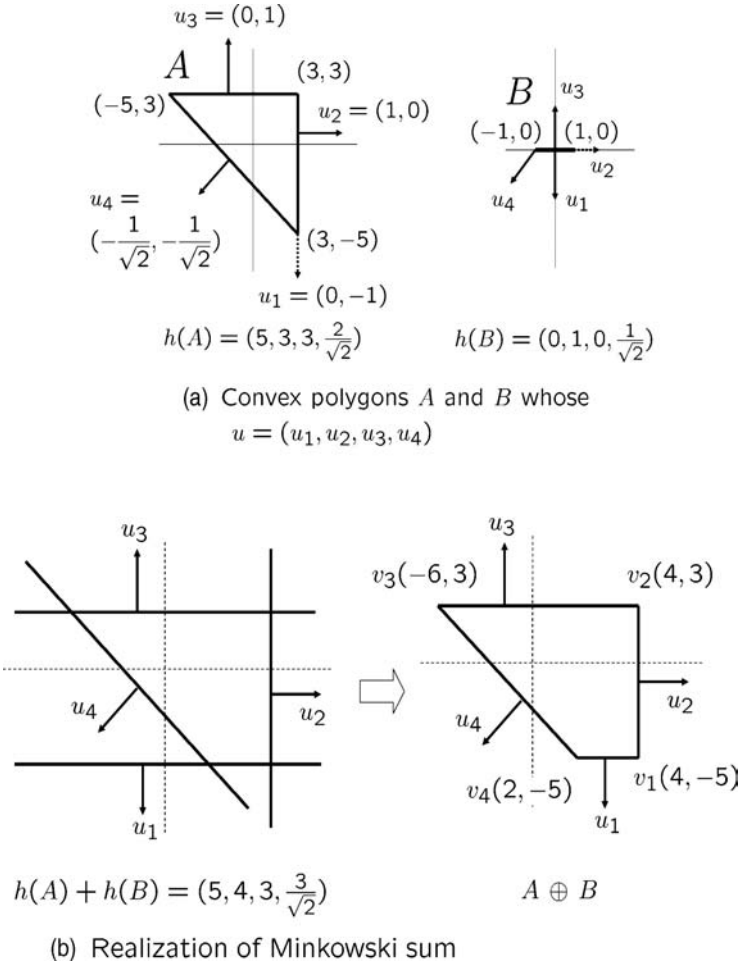


Figure 5.6 Minkowski addition by means of supporting function vectors: (a) convex polygons A and B whose $u = (u_1, u_2, u_3, u_4)$; (b) the realization of the Minkowski sum from its supporting vector specification in terms of intersection of half-spaces, and then conversion of the sum polygon in terms of vertices and edges

$A \ominus B$ can be obtained by discarding the redundant half-spaces specified by the vector $(\eta_1^A - \eta_1^B, \dots, \eta_n^A - \eta_n^B)$.

The novelty of our approach lies in our treatment of *not discarding* those redundant half-spaces, but retaining them as essentials.

We will show the principle of computation of the Minkowski decomposition by using the supporting function vectors in Figure 5.7. In computing the decomposition $A \ominus B$ as $h(A) + (-h(B)) = (\eta_1^A - \eta_1^B, \dots, \eta_n^A - \eta_n^B)$ (in Figure 5.7(a)), note that one of the half-planes – namely, the half-plane corresponding to the outer normal u_1 – is redundant. The normal

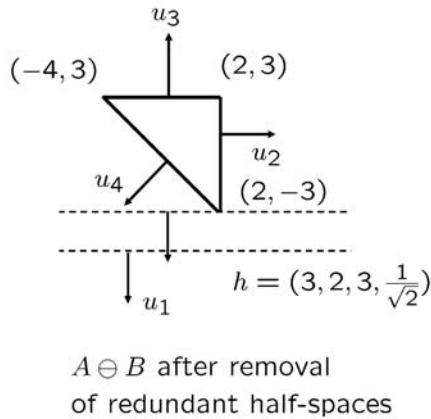
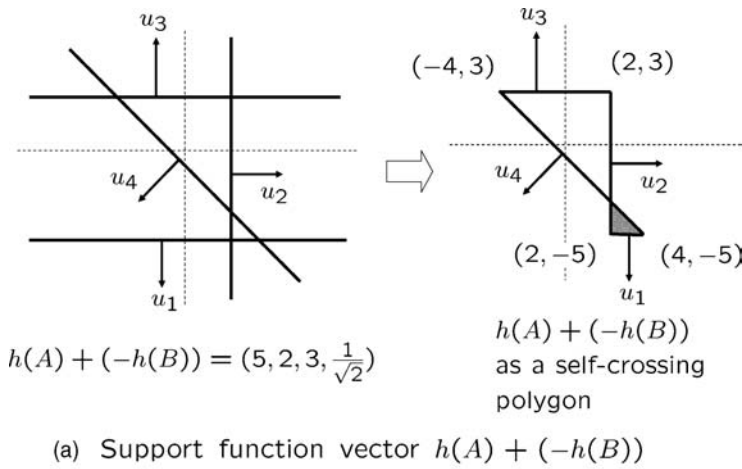


Figure 5.7 Minkowski decomposition by means of supporting function vectors: (a) the support function vector $h(A) + (-h(B))$ in terms of half-spaces, and then conversion of the half-spaces in terms of vertices and edges (which produces a self-crossing polygon); (b) the removal of redundant half-spaces – A and B are given in Figure 5.6

custom is to abandon this redundant half-plane altogether in further considerations. This effectively means an appropriate reduction of the corresponding value of the supporting function in the supporting function vector $h(A) + (-h(B))$. In our example, the corresponding supporting function value $\eta_1^A - \eta_1^B = 5$ has been reduced to 3 (see Figure 5.7(b)). This clearly explains why, in general, $(A \ominus B) \oplus B \subseteq A$.

5.5.2 Computation by Means of Edges: The Emergence of the Boundary Addition Operation \uplus

Representing a convex polygon A by means of its supporting function vector $h(A)$ is not a very common practice. A more popular representation is the *edge length* representation, where the boundary ∂A is represented by means of a starting vertex v_s^A and the length ι_i^A of the edge corresponding to every u_i ; that is, $\partial A = \{ \{v_s^A\}, \{\iota_1^A, \dots, \iota_n^A\} \}$. From the $h(A)$ -representation, it is easy to arrive at the edge length representation by finding the intersection points of adjacent half-spaces (see the right-hand plots in Figures 5.6(b) or 5.7(a)).

If the other summand B is also represented in a similar way; that is, $\partial B = \{ \{v_s^B\}, \{\iota_1^B, \dots, \iota_n^B\} \}$, then by a straightforward computation we get

$$h(A) + h(B) = \left\{ \{v_s^A + v_s^B\}, \{\iota_1^A + \iota_1^B, \dots, \iota_n^A + \iota_n^B\} \right\}. \quad (5.13)$$

Equation (5.13) states that the boundary $\partial(A \oplus B)$ of the Minkowski sum of two convex polygons can easily be computed when the boundaries ∂A and ∂B are represented in their edge length forms. The computation essentially involves *addition of the lengths of the corresponding edges of the summands*. This operation may be termed the boundary addition operation and denoted by the symbol “ \uplus .” Equation (5.13) can, therefore, be rewritten as

$$\begin{aligned} \partial(A \oplus B) &= \partial A \uplus \partial B \\ &= \left\{ \{v_s^A + v_s^B\}, \{\iota_1^A + \iota_1^B, \dots, \iota_n^A + \iota_n^B\} \right\}. \end{aligned} \quad (5.14)$$

For computing Minkowski decomposition, we convert the support function vector $-h(B)$ into the equivalent edge length representation, which turns out to be

$$-h(B) = \left\{ \{-v_s^B\}, \{-\iota_1^B, \dots, -\iota_n^B\} \right\}.$$

The polygon represented by $-h(B)$, if drawn pictorially, looks like a hole or a *negative region*, without any positive region surrounding the hole (Figure 5.8(b)). This is because, for any ordinary convex polygon (Figure 5.8(a)), all the outer normals u_i ’s “diverge outward,” whereas the normals u_i ’s of $-h(B)$ appear to “converge inward”; but the *directions* of the normals remain exactly the same in both cases. We can distinguish these two cases by introducing the concept of the “sense” of the outer normals. The outer normals of ordinary convex objects, which diverge outward, can be thought of as having a “positive” sense; whereas the normals of $-h(B)$ have a “negative” sense. We can consider the polygon represented by $-h(B)$ as the additive inverse of the polygon B , and we can denote it by the symbol B^{-1} .

As far as the geometric shape is concerned, B^{-1} appears exactly like the symmetrical set \check{B} . But note the differences among the objects B , \check{B} , and B^{-1} . The sense of every outer normal of both B and \check{B} is the same – that is, positive – while the directions of the outer normals at the corresponding faces of the two objects are exactly opposite. Because of the positive sense of the outer normals, we consider both B and \check{B} as positive objects. On the contrary, the directions of the outer normals at the corresponding faces of B and B^{-1} are exactly the same, but the senses are opposite. Because of the negative sense of the outer normals, B^{-1} can be considered as a negative object.

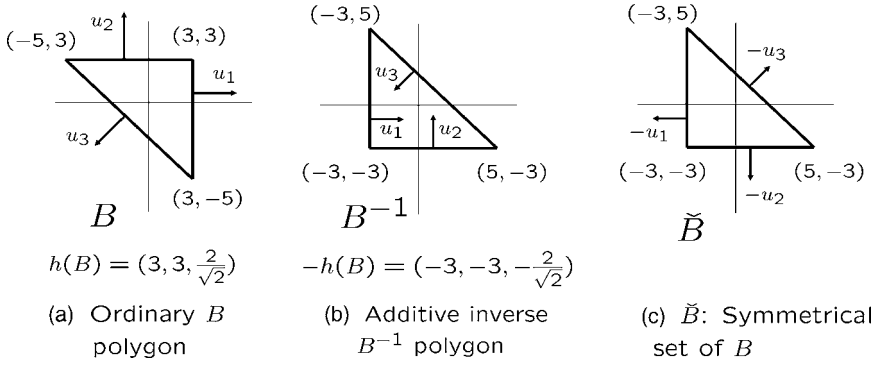


Figure 5.8 (a) The ordinary B polygon; (b) a geometric interpretation of the inverse shape B^{-1} ; and (c) for comparison, the symmetrical set of B , \tilde{B}

Using the notation of boundary addition, we can write

$$\begin{aligned}
 h(A) + (-h(B)) &\equiv \partial A \uplus \partial B^{-1} \\
 &= \left\{ \{v_s^A - v_s^B\}, \{\iota_1^A - \iota_1^B, \dots, \iota_n^A - \iota_n^B\} \right\}. \quad (5.15)
 \end{aligned}$$

Since $h(A) + (-h(B))$ may contain redundant half-spaces, the polygon represented by $\partial A \uplus \partial B^{-1}$ will be, in general, a self-crossing polygon (see the right-hand plot in Figure 5.7(a)). With regard to the senses of the outer normals, a self-crossing polygon may appear to have both a positive part and a negative part (the negative part is shown shaded in Figure 5.7(a)). Since we obtain $A \ominus B$ from $h(A) + (-h(B))$ by discarding the redundant half-spaces, equivalently $\partial(A \ominus B)$ is obtained from $\partial A \uplus \partial B^{-1}$ by discarding the negative part of the polygon $\partial A \uplus \partial B^{-1}$, and considering only its positive part. This can be expressed symbolically as

$$\partial(A \ominus B) = \text{Pos}(\partial A \uplus \partial B^{-1}), \quad (5.16)$$

where $\text{Pos}(X)$ denotes a unary operation, which extracts the positive portion of a self-crossing object X .

Note: In the case that we are only interested in the “shapes” of the objects, and not their positions in the plane, then the starting vertices v_s^A , v_s^B , and so on (which appeared in (5.14) and (5.15)) can be ignored.

5.5.3 Computation by Means of Slope Diagrams: The Unification of Minkowski Addition and Decomposition

The *slope diagram* representation of a polygon is essentially the edge length representation in a pictorial form. Since all the outer normals u_i ’s of a convex polygon lie on a unit circle S^1 , in slope diagram representation we consider a unit circle as the basis. The representation scheme is as follows (see Figure 5.9):

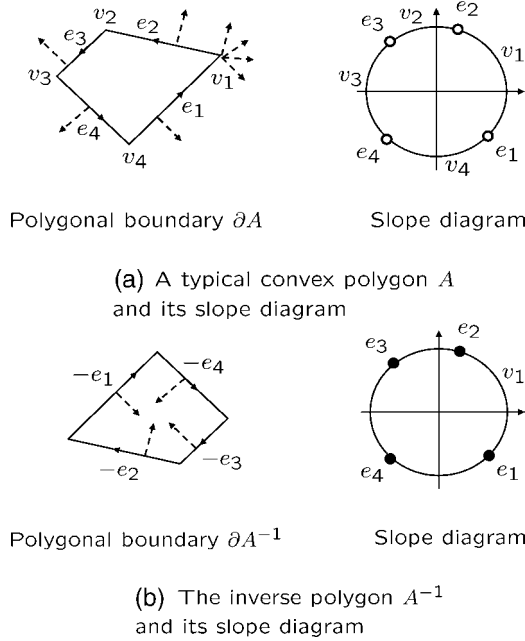


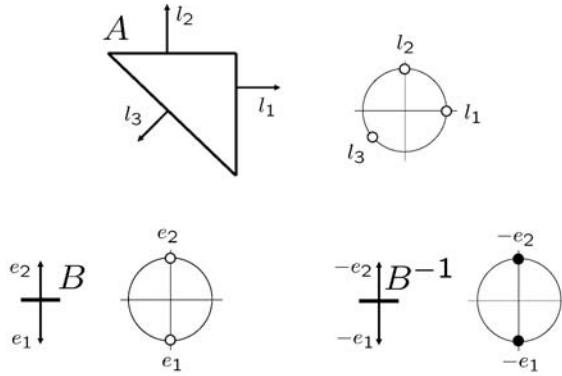
Figure 5.9 (a) A typical convex polygon A and its slope diagram; (b) the inverse polygon A^{-1} and its slope diagram

1. The outer normal direction at each edge of the polygon is represented by the corresponding point on a unit circle, which is called an *edge point*. (By “corresponding point,” we mean that point on the unit circle where the outer normal direction is the same as the outer normal direction of the edge.)
2. At each vertex of the polygon, it is possible to draw innumerable many outer normals filling an angle (supplementary to the interior angle at the vertex). This set of outer normal directions at the vertex is represented by the corresponding arc on the unit circle, which is called a *vertex arc*.
3. Apart from the direction of an outer normal, the sense of the outer normal must also be indicated. If the sense of an outer normal is negative, it will be shown by heavy black points or arcs, while an outer normal that has a positive sense will be drawn using light lines.
4. The length of each edge is associated with its corresponding edge point, like a label.

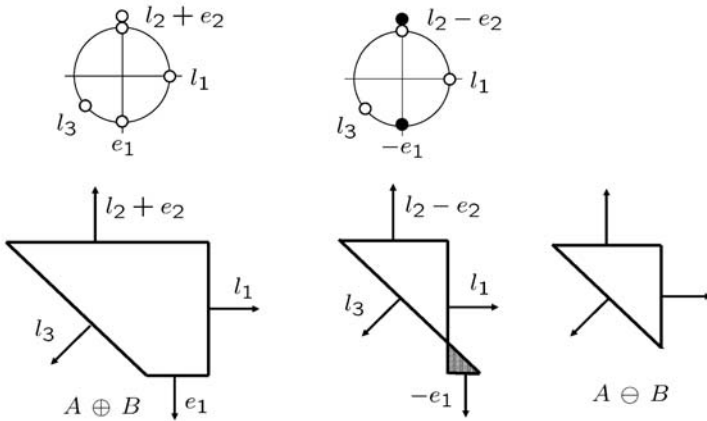
5.5.4 The Computation of Boundary Addition

The computation of $\partial A \uplus \partial B$ or $\partial A \uplus \partial B^{-1}$ (see (5.14) or (5.15); we will ignore v_s^A, v_s^B , etc. at present) by means of slope diagrams becomes quite straightforward:

1. *Merging of slope diagrams.* Merge the slope diagrams of the two operands (given in Figure 5.10(a)) into a single one.



(a) Operands A, B and its additive inverse, and the corresponding slope diagrams



(b) Merged slope diagram of A and B and realization of $\partial A \uplus \partial B$

(c) Merged slope diagram of A, B^{-1} and $\partial A \uplus \partial B^{-1}$, and Minkowski decomposition $A \ominus B$

Figure 5.10 Minkowski addition and decomposition by means of slope diagrams – in this case, $\partial A \uplus \partial B = \partial(A \oplus B)$, but $\partial(A \ominus B) = \text{Pos}(\partial A \uplus \partial B^{-1})$: (a) the operands A, B , and the additive inverse B^{-1} , and the corresponding slope diagrams; (b) the merged slope diagram of A and B and the realization of $\partial A \uplus \partial B$; (c) the merged slope diagram of A, B^{-1} , and $\partial A \uplus \partial B^{-1}$, and Minkowski decomposition $A \ominus B$

2. *Realization of the boundary sum.* From the merged slope diagram, realize the polygon that it represents. The term “realization” means “concatenation” of the edges (that is, joining the end-point of one edge to the start-point of the next edge) in the sequence in which they appear in the merged slope diagram (Figures 5.10(b) or (c)).

Another example of Minkowski addition and decomposition by means of slope diagrams is given in Figure 5.11. The realization/concatenation process may need some further clarification.

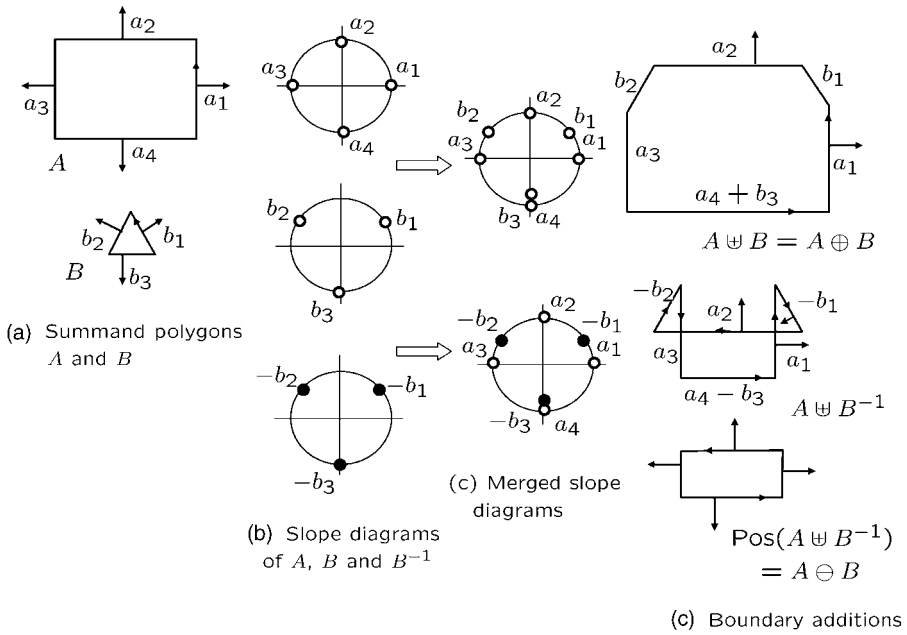


Figure 5.11 Another example of Minkowski addition and decomposition by means of slope diagrams: (a) summand polygons A and B ; (b) the slope diagrams of A , B , and B^{-1} ; (c) the merged slope diagrams; (d) the boundary additions

First, see (5.14), concerning Minkowski addition. In the merged slope diagram, two distinct cases may arise: (1) If two edge points of the summands occur at the same position of the unit circle, this means that both of the edges have the same outer normal direction. Therefore, concatenation of these two edges effectively means the addition of the lengths $\iota_i^A + \iota_i^B$ of the edges. (2) If one edge point of a summand lies on a vertex arc of the other summand, this means that the edge of the second summand has zero length at that outer normal direction. The addition of zero length simply involves consideration of the edge of the first summand, and that is precisely what we achieve by concatenation. Next, see (5.15), concerning Minkowski decomposition. Here, everything remains as before, except that some of the edge points (resulting from the B^{-1} polygon) in the merged slope diagram are black; that is, have negative sense. If the sense is negative, we have to subtract the corresponding length of the edge of B from that of A . Clearly, the subtraction of one directed edge from another simply involves reversing the direction of the former and then concatenating with the latter.

This slope diagrammatic approach clearly illustrates the following:

- Just as in real number arithmetic, instead of treating the division m/n as a separate operation, we can consider it as multiplication with the multiplicative inverse $\frac{1}{n}$, in a similar way, the Minkowski decomposition $\partial(A \ominus B)$ operation can also be viewed as addition with the additive inverse ∂B^{-1} .

- Just as in the system $(\mathbb{N}, \cdot, /)$, we do not obtain *closure* under the division operation. Therefore, we need to extend the number domain from \mathbb{N} to the set of all positive rational numbers \mathbb{Q}^+ for the purpose of closure; and, similarly, to obtain closure under Minkowski decomposition it is necessary to extend the domain \mathcal{K} of the ordinary convex objects – it is done here by introducing the concept of negative shape.
- Both Minkowski addition and decomposition can be reduced to a single operation \uplus : Minkowski addition involves boundary addition \uplus of two positive (ordinary) objects, while Minkowski decomposition involves boundary addition of one positive and one negative object. Boundary addition, as we have shown, is essentially just addition of real numbers (representing the lengths of the edges of the operand polygons; those lengths may be positive or negative).
- Just as $\lfloor m/n \rfloor$ is obtained by discarding the fractional part of m/n , similarly $\partial(A \ominus B)$ is obtained by discarding the negative part of $\partial A \uplus \partial B^{-1}$.

5.6 In the Domain of Convex Polyhedra

One of the basic theses of this chapter is that Minkowski operations on three- or higher-dimensional boundary-represented objects in E^d eventually reduce to Minkowski operations on polygons in E^2 , which, in turn, boil down to addition of real numbers. At present, we shall consider the case of polyhedral objects, though our approach is general enough to extend beyond three dimensions. The approach is to find relations for convex polyhedra that will be similar to (5.14) and (5.15), and then to resort to the slope diagrammatic technique, as we have done for convex polygons.

5.6.1 Computation by Means of Faces

The boundary of a convex polyhedron can be represented by means of vertices, edges, and facets. Our approach demands that these concepts are defined more precisely.

If, for some $u \neq 0$, we have $H(A, u) < \infty$ (this condition ensures that A is bounded), then the hyperplane

$$L(A, u) = \{ p \in E^d \mid (p, u) = H(A, u) \}$$

is called the supporting hyperplane of A with *outer normal* u . In E^2 , the supporting hyperplane becomes the supporting line of a convex polygon A with outer normal u , while in E^3 it is the supporting plane of a convex polyhedron A .

A face of A with outer normal u , denoted by $F(A, u)$, is then defined as

$$F(A, u) = L(A, u) \cap A.$$

Thus $F(A, u)$ is precisely that set of boundary points of A where the outer normal is either u or parallel to u . Now, considering all the directions of the E^d space as the directions of the outer normal u , the collection of the corresponding faces will describe the entire boundary of A . That is, the entire boundary of A can be described as $\partial A = \bigcup_{u \in S^{d-1}} F(A, u)$.

This notion of $F(A, u)$ is related to our conventional concept of the boundary of an object in terms of vertices, edges, faces, and so on in the following way. If A is a convex d -dimensional object, then $L(A, u)$ is a $(d - 1)$ -dimensional hyperplane. Therefore, $F(A, u)$ may have

dimensions $0, 1, \dots, (d - 1)$. Normally, a face $F(A, u)$ of dimension r ($r = 0, 1, \dots, (d - 1)$) is called a r -face of A . A maximal proper face of A – that is, a $(d - 1)$ -dimensional face – is called a *facet* of A . Clearly, if A is a two-dimensional convex polygon, then $F(A, u)$ is either a 0-face (*vertex*) or an 1-face (*edge*). Since an edge of a polygon is a maximal proper face, it can also be called a facet of the polygon. When A is a three-dimensional convex polyhedron, in addition to being either a vertex or an edge, $F(A, u)$ may also be a 2-face (*facet*). (Note that, for three-dimensional objects, instead of “facet,” the term planar “face” is more frequently used.)

From Theorem 5.5, it is not difficult to derive the following result [34, 50].

Theorem 5.8. *Let A and B be two convex polytopes in E^d . Then, for every $u \in S^{d-1}$,*

$$F(A \oplus B, u) = F(A, u) \oplus F(B, u). \quad (5.17)$$

Therefore, in terms of faces of the summands, the boundary $\partial(A \oplus B)$ can be expressed as

$$\begin{aligned} \partial(A \oplus B) &= \bigcup_{u \in S^{d-1}} F(A \oplus B, u) \\ &= \bigcup_{u \in S^{d-1}} (F(A, u) \oplus F(B, u)). \end{aligned} \quad (5.18)$$

Equation (5.18) will be our basis for computing Minkowski operations on convex polyhedra. We shall first argue that it is not necessary to compute $(F(A, u) \oplus F(B, u))$ for every $u \in S^2$ in the three-dimensional space. Just as a convex polygon is completely specified by its oriented edges, similarly a convex polyhedron is completely specified by its oriented facets (planar faces). Therefore, it is sufficient if we compute only the facets of $A \oplus B$, but not every $F(A \oplus B, u)$'s. As is evident from (5.18), the facets of $A \oplus B$ can be obtained in the following way:

1. *Minkowski addition of two facets.* Adding a facet of A to a facet of B ; that is, $\text{facet}_A \oplus \text{facet}_B$.
2. *Minkowski addition of a facet and an edge.* Adding a facet of one of the two summands to an edge of the other; that is, $\text{facet}_A \oplus \text{edge}_B$ or $\text{edge}_A \oplus \text{facet}_B$.
3. *Minkowski addition of a facet and a vertex.* Adding a facet of one of the two summands to a vertex of the other; that is, $\text{facet}_A \oplus \text{vertex}_B$ or $\text{vertex}_A \oplus \text{facet}_B$.
4. *Minkowski addition of two nonparallel edges.* Adding nonparallel edges of A and B ; that is, $\text{edge}_A \oplus \text{edge}_B$.

Here, the facets, edges, and vertices added in this way lie in supporting planes, with parallel outer normals.

Now, Minkowski addition of two facets that lie in supporting planes having parallel outer normals is equivalent to Minkowski addition of those facets lying in the same plane. That, in turn, is simply Minkowski addition of two convex polygons in E^2 . The same is true for Minkowski addition of a facet and an edge, or the addition of two edges.

We depict a typical such addition in Figure 5.12. (In a similar way, we can continue further and show that Minkowski addition of two convex polytopes in E^d reduces to Minkowski addition operations in E^{d-1} , \dots , and eventually reduces to the Minkowski addition of convex polygons in E^2 . This means that it finally reduces to addition of real numbers.)

In computing the Minkowski decomposition $A \ominus B$, exactly as in the polygonal case, the first step is to compute the boundary addition $\partial A \uplus \partial B^{-1}$, and then to determine its positive

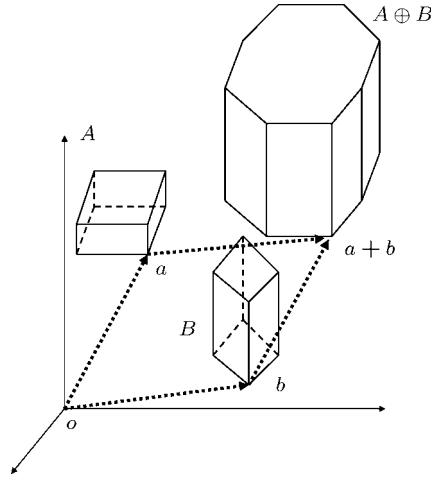


Figure 5.12 The computation of $\bigcup_{u \in S^{d-1}} (F(A, u) \oplus F(B, u))$ for two convex polyhedra A and B in E^3

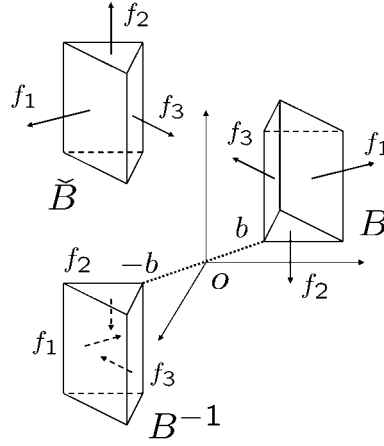
portion $\text{Pos}(\partial A \uplus \partial B^{-1})$. In terms of the faces of the operand polyhedra, the first step is to compute $\bigcup_{u \in S^{d-1}} (F(A, u) \oplus F(B^{-1}, u))$. Therefore, according to our previous discussion, it will reduce to $\text{facet}_A \oplus \text{facet}_{B^{-1}}$, $\text{facet}_A \oplus \text{edge}_{B^{-1}}$ or $\text{edge}_A \oplus \text{facet}_{B^{-1}}$, $\text{facet}_A \oplus \text{vertex}_{B^{-1}}$ or $\text{vertex}_A \oplus \text{facet}_{B^{-1}}$, and $\text{edge}_A \oplus \text{edge}_{B^{-1}}$ (where they lie in supporting planes with parallel outer normals). As we have argued previously, each of these additions can first be reduced to the boundary addition of a positive and a negative convex polygon in E^2 , which, in turn, reduces to subtraction of real numbers.

We present an example of a negative polyhedron, B^{-1} , in Figure 5.13(a). In Figure 5.13(b), we show the Minkowski decomposition of two convex polyhedra by means of boundary addition \uplus and, thereafter, the Pos operation.

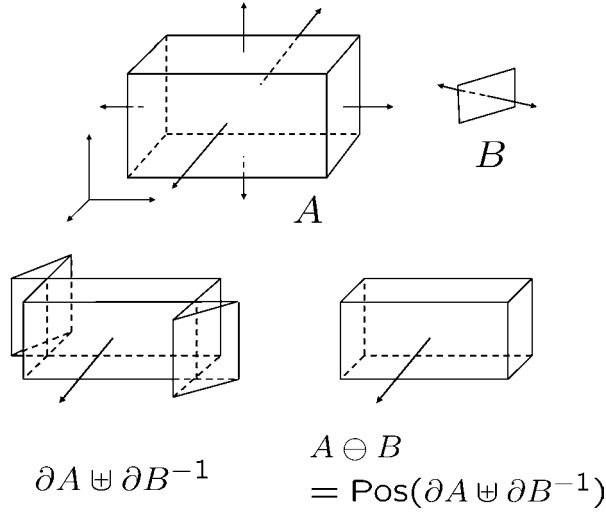
5.6.2 The Slope Diagram Representation of a Convex Polyhedron

The essential goal of the slope diagram representation of an object is to capture the behavior of the outer normals of the object in an explicit way. The behavior of the directions of the outer normals at various faces (that is, at the facets, edges, and vertices) of a convex polyhedron can be described as follows:

1. At each interior point of a facet of a polyhedron, there can be drawn only one outer normal.
2. At each point of an edge (different from a vertex), it is possible to draw an infinite number of normals filling a plane angle, which will be supplementary to the corresponding interior angle. (In some of the literature, this interior angle is called the *dihedral angle* corresponding to the edge. The exact definition is as follows: the dihedral angle corresponding to an edge is the angle between the planes of its two adjacent facets.)
3. At each vertex, it is possible to draw infinitely many outer normals that fill a solid angle (interpolating between the incident facet normals).



(a) Polyhedron B and corresponding B^{-1} and \check{B} polyhedra



(b) The input A and B are convex polyhedra

Figure 5.13 (a) The geometric representation of a negative polyhedron B^{-1} – polyhedron B and the corresponding B^{-1} and \check{B} polyhedra; (b) the computation of $A \ominus B$ using the boundary addition operation – the input A and B are convex polyhedra

For the slope diagram of a polyhedron, we have to start with a unit sphere S^2 .

- (a) *Facet representation.* As in the polygonal case, each facet can be represented by the corresponding point on the unit sphere, which is referred to as a *facet point*.

- (b) *Edge representation.* Each edge of the polyhedron, we claim, can be represented by the arc of the great circle joining the two facet points corresponding to the two adjacent facets of the edge. We call such an arc an *edge arc* of the polyhedron.

Note: The intersection of the surface of a sphere by a plane is called a great circle if the plane passes through the center of the sphere. Clearly, only one great circle can be drawn through two given points on the surface of the sphere, except when the points are the extremities of a diameter of the sphere. By the “arc of a great circle,” we generally mean the shorter of the two arcs joining the two points.

- (c) *Vertex representation.* According to the scheme, it easy to see that the directions of the outer normals at any vertex v of the polyhedron will be represented by a region on the unit sphere. This region is bounded by the edge arcs corresponding to the edges incident at v , and the vertices of this region are the facet points corresponding to the facets of the polyhedron incident at v . We call this a *vertex region*.
- (d) To denote the sense of an outer normal and the length of an edge, we use the same conventions as adopted for polygons.

In Figure 5.14, we show the slope diagram representations of two convex polyhedra. The heavy lines and solid black dots in the diagrams indicate the negative sense of the outer normals.

Note: In Mount and Silverman [75] and Guibas and Seidel [36], we find notions that are similar in some ways to the idea of the slope diagram of a convex polyhedron. However, these authors did not take into account the notion of “sense” of an outer normal, which is crucial in our approach.

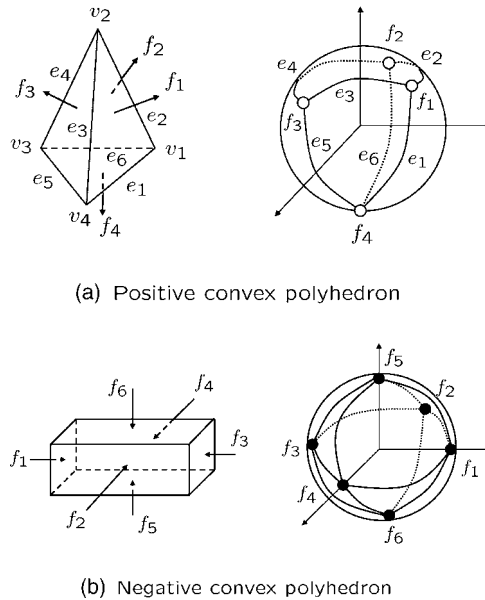


Figure 5.14 Two convex polyhedra and their slope diagram representations (not to scale): (a) a positive convex polyhedron; (b) a negative convex polyhedron

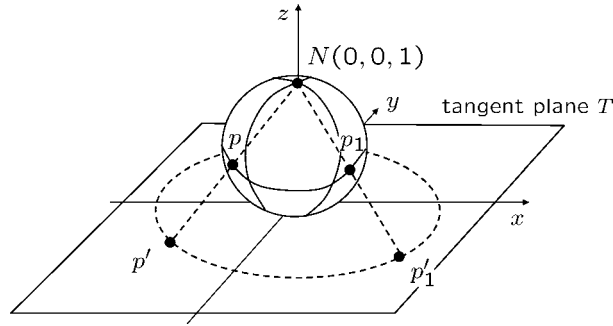


Figure 5.15 Transforming a three-dimensional slope diagram by stereographic projection on two dimensions

Since the slope diagram of a polyhedron is a three-dimensional figure, it may not be a very convenient representation to deal with for the purpose of intuitive understanding and visualization. Therefore, we transform this representation into an equivalent two-dimensional form by means of the *stereographic projection*. As shown in Figure 5.15, the stereographic projection is a projection of a sphere from one of the points N onto the plane T tangent to the sphere at the diametrically opposite point N' . The point N is called the *projection center*.

The transformation equations for the stereographic projection are given in many standard texts, such as [88]. Consider a unit sphere whose center o is at the origin, and the projection center N for which is located on the $o - z$ axis (see Figure 5.15). In this case, the point N has the coordinates $(0, 0, 1)$ and the projection plane T is the plane $z = -1$. Let the point $p(x, y, z)$ of the sphere be projected stereographically into the point $p'(x', y', -1)$ of the plane T . Then the coordinates of the point p , corresponding to the point p' , are equal to

$$\begin{aligned} x &= \frac{4x'}{x'^2 + y'^2 + 4}, \\ y &= \frac{4y'}{x'^2 + y'^2 + 4}, \\ z &= \frac{x'^2 + y'^2 - 4}{x'^2 + y'^2 + 4}. \end{aligned}$$

The inverse mapping – that is, the coordinates of the point p' , corresponding to the point p – is equal to

$$x' = \frac{2x}{1 - z}, \quad y' = \frac{2y}{1 - z}, \quad z' = -1.$$

From the above equations, we find that if p is the projection center itself – that is, $p = (0, 0, 1)$ – then we have difficulty in computing x' and y' . This difficulty is circumvented by extending the plane T by an “ideal” point, which is called the *point at infinity*.

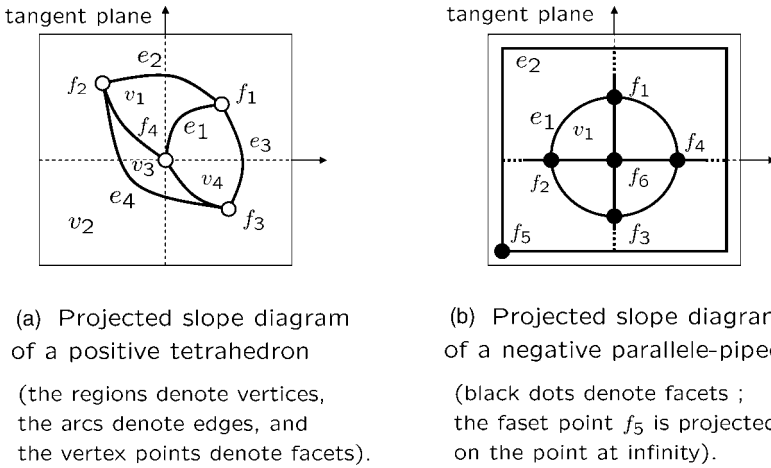


Figure 5.16 Stereographic projections of the slope diagrams of the convex polyhedra depicted in Figure 5.14 (not to scale): (a) the projected slope diagram of a positive tetrahedron (the regions denote vertices, the arcs denote edges, and the vertex points denote facets); (b) the projected slope diagram of a negative parallelepiped (black dots denote facets – the facet point f_5 is projected on the point at infinity)

In the stereographic projection of a slope diagram, any facet point at (x_1, y_1, z_1) will be projected onto the point (x'_1, y'_1) of the T -plane; the point (x'_1, y'_1) can be determined using the above equations. Any edge arc will be projected either as a straight line or a circular arc. An edge arc joining two facet points (x_1, y_1, z_1) and (x_2, y_2, z_2) will be projected on the plane T as $(x_1 y_2 - x_2 y_1)(x'^2 + y'^2) + 4(y_1 z_2 - y_2 z_1)x' + 4(z_1 x_2 - z_2 x_1)y' - 4(x_1 y_2 - x_2 y_1) = 0$. This is the equation of a circular arc – unless $x_1 y_2 = x_2 y_1$, when it degenerates into a straight-line segment.

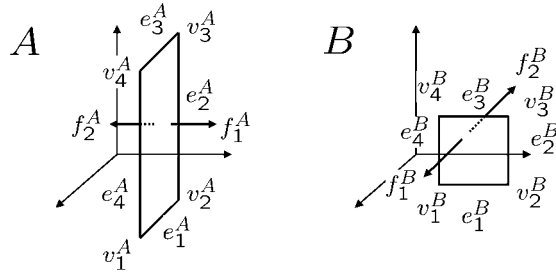
In Figure 5.16, we present the stereographic projections of the slope diagrams shown in Figure 5.14.

5.6.3 Computation by Means of Slope Diagrams

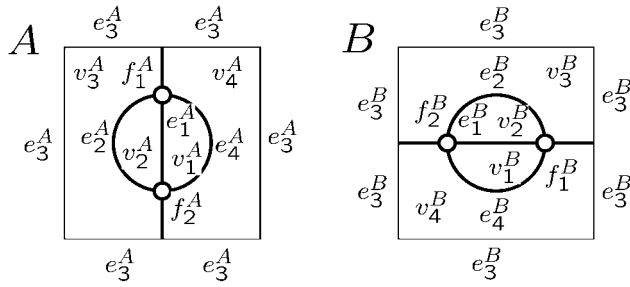
Assume that two convex polyhedra, A and B , are given in the form of their slope diagram representations. Exactly as in the polygonal case, the boundary addition computation involves two steps: (a) merging of slope diagrams of the operands, and (b) realization of the boundary sum from the merged slope diagram.

If we *merge* these two slope diagrams (that is, overlay both of the slope diagrams on the same unit sphere S^2), we can immediately identify the corresponding $F(A, u)$'s and $F(B, u)$'s (or $F(B^{-1}, u)$'s in the case of decomposition) that have the same outer normal direction u , because they will occupy the same positions on the sphere. This means that *wherever there are intersections between two slope diagrams, the corresponding faces of the operands need to be added (or subtracted)*.

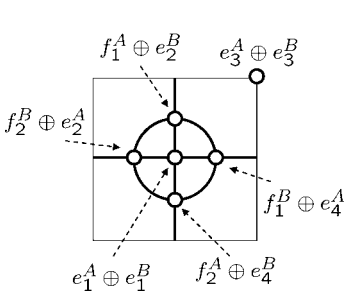
To *realize* the boundary sum $\partial A \uplus \partial B$ or $\partial A \uplus \partial B^{-1}$ from the merged slope diagram, we have to identify only the facet points of the boundary sum. Any facet point will be created in one of the following ways: (a) the intersection of a facet point of one operand with that of the



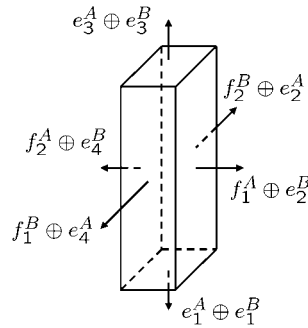
(a) Operands A and B ; facets edges, and vertices are denoted by f 's, e 's, v 's respectively.



(b) Slope diagram representations of the operands.



(c) Merged slope diagram:
two new face points are created
by non-parallel edges, i.e.,
 $e_1^A \oplus e_1^B$ and $e_3^A \oplus e_3^B$.



(d) $\partial A \uplus \partial B$ since
both A and B are positive
convex polyhedra, and
 $\partial(A \oplus B) = \partial A \uplus \partial B$.

Figure 5.17 The Minkowski addition of two polyhedra using their slope diagram representations: (a) the operands A and B – the facets, edges, and vertices are denoted by the f 's, e 's, and v 's, respectively; (b) the slope diagram representations of the operands; (c) the merged slope diagram, in which two new faces are created by nonparallel edges – that is, $e_1^A \oplus e_1^B$ and $e_3^A \oplus e_3^B$; (d) $\partial A \uplus \partial B$, since both A and B are convex polyhedra, and $\partial(A \oplus B) = \partial A \uplus \partial B$.

other, which means *addition of two facets*; (b) the intersection of a facet point of one operand with an edge arc of the other, which means *addition of a facet and an edge*; (c) the intersection of a facet point of one operand with a vertex region of the other, which means, *addition of a facet and a vertex*; or (d) the intersection of an edge arc of one operand with a nonparallel edge arc of the other, which means *addition of two nonparallel edges*.

In the case of Minkowski decomposition, if $\partial A \uplus \partial B^{-1}$ turns out to be a self-crossing polyhedron (as shown in Figure 5.13(b)), we have to discard the negative portion and consider only the positive portion.

In Figure 5.17, we demonstrate computation by means of slope diagrams through an example of addition of a rectangular plane (A) that is parallel to the xz -plane to another rectangular plane (B) parallel to the yz -plane. The resulting sum shape $A \oplus B$ will be a rectangular parallelepiped.

6

Morphological Operations on Nonconvex Objects

6.1 Problems with Nonconvex Objects

If the summands A and B in $A \oplus B$ are nonconvex objects, Theorem 5.5, in the previous chapter, which tells us that

$$H(A \oplus B, u) = H(A, u) + H(B, u), \quad (6.1)$$

and

$$F(A \oplus B, u) = F(A, u) \oplus F(B, u) \quad (6.2)$$

for every $u \in E^d$, no longer holds. We then encounter three kinds of problem. In this chapter, we briefly discuss those problems and suggest some remedies, so that Minkowski operations on nonconvex as well as convex objects can be viewed through a single algorithmic framework. (Unless otherwise stated, the operands are assumed to be simply connected.)

6.1.1 A Localized Definition of $F(A, u)$

Unlike a convex object, a hyperplane $L(A, u)$ supporting a nonconvex object A at some point a on the boundary of ∂A may not be a supporting hyperplane of A – it may intersect the interior of A as shown in Figure 6.1.

To remedy this situation, we can extend the notion of supporting hyperplane to *relative/local supporting hyperplane*. The idea is to consider not the whole of the object A , but only a *neighborhood* of the point a . The neighborhood of a , a very small circular disk (or, in the case of a three-dimensional object, a spherical ball), is commonly denoted by $N(a)$. We call $L(A, u)$ a local supporting hyperplane of A if it is a supporting hyperplane of the set $A \cap N(a)$. This means that $L(A, u)$ is a supporting hyperplane only locally near the point $a \in \partial A$, but not globally for all the points of ∂A . Assuming $L(A, u)$ to be the local supporting hyperplane, we can extend the definition of a face $F(A, u)$ accordingly; that is, $F(A, u) = L(A, u) \cap (A \cap N(a))$.

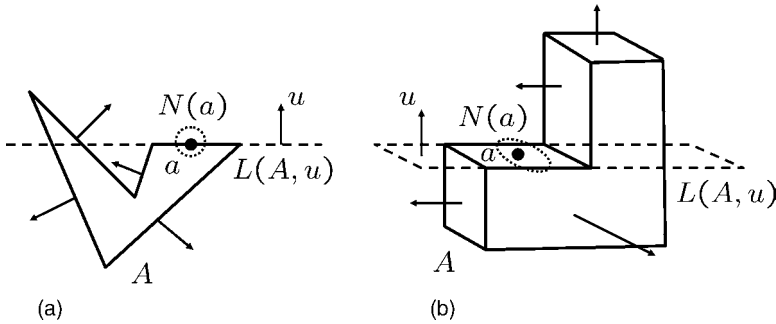


Figure 6.1 The local supporting hyperplane $L(A, u)$ in the neighborhood of a point a of A : (a) nonconvex polygon; (b) nonconvex polyhedron

The price that we pay for this localized definition of $F(A, u)$ is as follows: if any of the summands A or B is nonconvex, it can no longer be assumed, as in the convex case, that all the points in the collection $\bigcup_{u \in S^{d-1}} (F(A, u) \oplus F(B, u))$ will lie on the global boundary $\partial(A \oplus B)$ of the sum; some of the points in the collection may happen to be interior points of $A \oplus B$. Symbolically, $\partial(A \oplus B) \subseteq \bigcup_{u \in S^{d-1}} (F(A, u) \oplus F(B, u))$.

6.1.2 The Anomalous Behavior of the Outer Normals at the Nonconvex Faces

Some of the faces of a nonconvex object are convex faces, while the rest are nonconvex faces. In Figure 6.2(a) the vertex v_2 , or the edges a_1, a_2 of the polygon, are nonconvex faces, while v_1, v_3 , or edges a_3, a_4 , are convex; similarly, the edge e_2 of the polyhedron of Figure 6.2(b) is

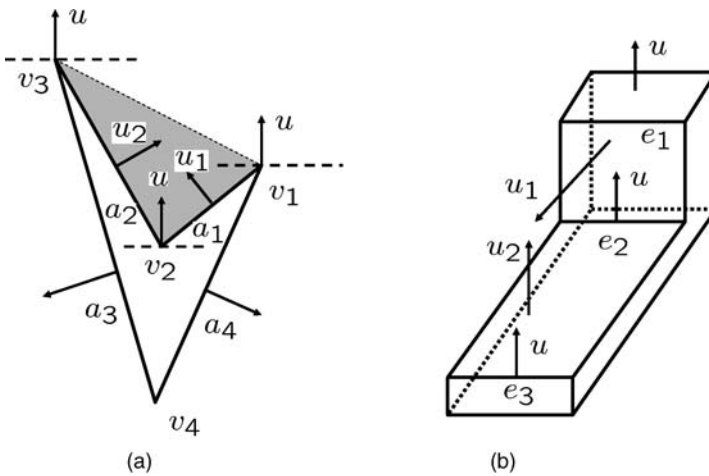


Figure 6.2 The anomalous behavior of the outer normals at the nonconvex faces: (a) nonconvex polygon; (b) nonconvex polyhedron

a nonconvex face. (The term “nonconvex face” is used here in the following sense. Consider a simple (nonconvex) polygon. A vertex of this polygon is called a nonconvex vertex (v_2 in the example figure) if the internal angle at the vertex is more than 180° ; otherwise, it is convex. The edges of the polygon that are incident to a nonconvex vertex (a_1 and a_2 in Figure 6.2(a)) are called nonconvex edges – or, in general, nonconvex faces. Similarly, for a polyhedron an edge is nonconvex if the internal angle is more than 180° , and the facets incident to a nonconvex edge are the nonconvex faces. This case is shown in Figure 6.2(b).)

The basic problem is that *the nature of an outer normal at a nonconvex face is different from that at a convex face*. As we have already described in Section 5.5.2, all of the faces of an ordinary convex object are convex, and the outer normals at any two adjacent convex faces appear to diverge outward from a point inside the object. In contrast to this, the outer normals at any two adjacent nonconvex faces converge to a point outside the object.

We remedy this problem in the following way. Consider the complementary region of the nonconvex part of the object A (part of this complementary region for the polygon is shown shaded in Figure 6.2(a)). This complementary region is like a hole or a negative region, and the nonconvex faces of A constitute a part of the boundary of this hole. (Intuitively, *a nonconvex object can be regarded as a combination of positive and negative convex objects*).

If we had added (in the Minkowski addition sense) the positive object B to this hole, the resulting faces would have been determined by subtracting each face of the hole from the corresponding face of B . In other words, a nonconvex face $F_2(A, u)$ of A and the corresponding convex face $F(B, u)$ of B are of opposite types; if one is considered to be positive, the other one will be considered to be negative. Since we are adding to B not the hole, but the positive part of A , this necessitates that the corresponding face of B has to be subtracted from the nonconvex face of A .

Following the same logic, in computing $A \ominus B$, which reduces to the boundary addition of the positive object A and the negative object B^{-1} , a convex face of B^{-1} is subtracted from the corresponding convex face of A , whereas it has to be added to the corresponding nonconvex face of A .

6.1.3 The Need to Maintain Explicit Topological Information about the Operands

Here, by “topological information” we mean how the various faces of the object are connected. In the case of a convex polytope, the topological information need not be maintained explicitly, since it can be easily derived from the outer normal directions of the faces. For a given outer normal direction u , a convex polytope has one and only one face; moreover, the faces of a convex polytope are connected in such a way that their outer normal directions are automatically arranged in sorted angular order (although the sorting orders for three- and higher-dimensional polytopes are more complicated). For this reason, we have already seen, in the case of a convex polygon, that the topological connections are automatically established if the consecutive edge points and vertex arcs are appropriately marked on the unit circle of its slope diagram (see Figure 6.3(a), where we can say, by inspecting the slope diagram of a polygon, that an edge e_1 is connected to the edge e_2 , etc.).

However, this does not happen with nonconvex objects. Consider the nonconvex polygon shown in Figure 6.2(a). For some outer normal direction, say u , the polygon has three vertices,

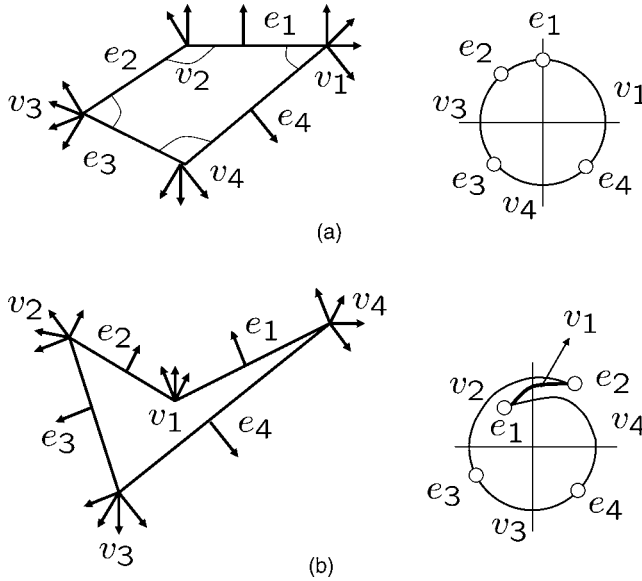


Figure 6.3 Slope diagram representations of a convex polygon and a nonconvex polygon: (a) a typical convex polygon and its slope diagram representation; (b) the slope diagram representation of a nonconvex polygon (the dark arc indicates the negative sense of the corresponding vertex)

v_1 , v_2 , and v_3 , and the vertices are all disconnected. The remedy is to maintain explicitly, in case of a nonconvex object, the information on how the various faces of the object are connected. As we shall show shortly, this can be easily achieved by means of the slope diagram representation.

6.2 Slope Diagrams for Nonconvex Polygons

6.2.1 The Boundary Addition of Nonconvex Polygons by Means of Slope Diagrams

The slope diagram representation of a nonconvex polygon is slightly more complicated than that of a convex polygon. For the convex edges and vertices, the representation is exactly the same as explained in Section 5.5.3. For the nonconvex portion, the following additional considerations must be taken into account (see Figure 6.3):

- To maintain the topological connectivity of the edges, we have to observe a forward and backward motion along the unit circle corresponding to a nonconvex vertex (Figures 6.4(b) or 6.5(b)).
- The *sense* of the outer normals at a nonconvex vertex is opposite to that at a convex vertex. Therefore, the vertex arc corresponding to the nonconvex vertex must be depicted by heavy black lines, to indicate its negative sense.

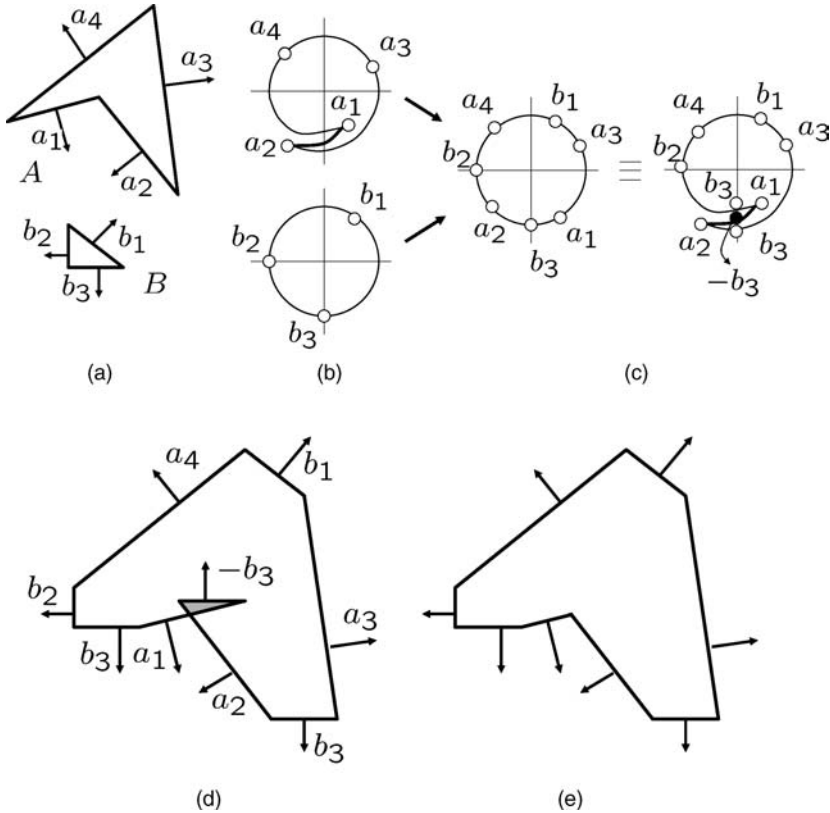


Figure 6.4 The Minkowski addition of nonconvex polygons by means of slope diagrams: (a) operand polygons A and B; (b) slope diagram representations of A and B; (c) the merged slope diagram; (d) $\partial A \oplus \partial B$; (e) $\partial(A \ominus B) = \text{Pos}(\partial A \oplus \partial B)$

Computation of boundary addition \oplus by means of slope diagrams remains exactly the same; that is, (a) merging of the slope diagrams of the operands, and (b) realization of the boundary sum from the merged slope diagram. However, the second step is slightly more involved, just in case any of the operands is nonconvex.

First, consider the boundary addition $\partial A \oplus \partial B$. As we have noted, in a nonconvex portion of a slope diagram, the path along the unit circle is traversed “three times” – twice in the positive sense and once in the negative sense (Figures 6.4(b) or 6.5(b)). Therefore, at the time of realization of the merged slope diagram, if there is any edge point of the other summand lying within this portion, it must be considered three times in the appropriate manner. The term “appropriate manner” means that in the negative arc portion (depicted by heavy black lines) the edge has to be subtracted, while in the positive arc portions it has to be added in the usual way (Figure 6.4(c)). Clearly, the subtraction of one directed edge from another simply involves reversing the direction of the former and then adding it to the latter (Figure 6.4(d)).

One important point to be noted here is the following. Since a nonconvex polygon is treated here as a combination of a positive object and a negative object, the boundary addition $\partial A \uplus \partial B$ of nonconvex polygons, unlike in the convex case, does not directly produce the boundary $\partial(A \oplus B)$ of the Minkowski sum. In general, as shown in Figure 6.4(d), $\partial A \uplus \partial B$ may be an oriented self-crossing polygon. The boundary of the sum can be obtained by determining the positive region enclosed by the resulting self-crossing polygon (Figure 6.4(e)). We can symbolically express the complete Minkowski addition as

$$\partial(A \oplus B) = \text{Pos}(\partial A \uplus \partial B) \quad (6.3)$$

for general polygons – convex or nonconvex.

The computation of $\partial A \uplus \partial B^{-1}$ is in no way different, except that the senses of the edge points and the vertex arcs of B^{-1} are exactly opposite to those of B . $\partial A \uplus \partial B^{-1}$ will be, in general, a self-crossing polygon, and $\partial(A \ominus B) = \text{Pos}(\partial A \uplus \partial B^{-1})$.

In Figure 6.5, we show an example of Minkowski decomposition.

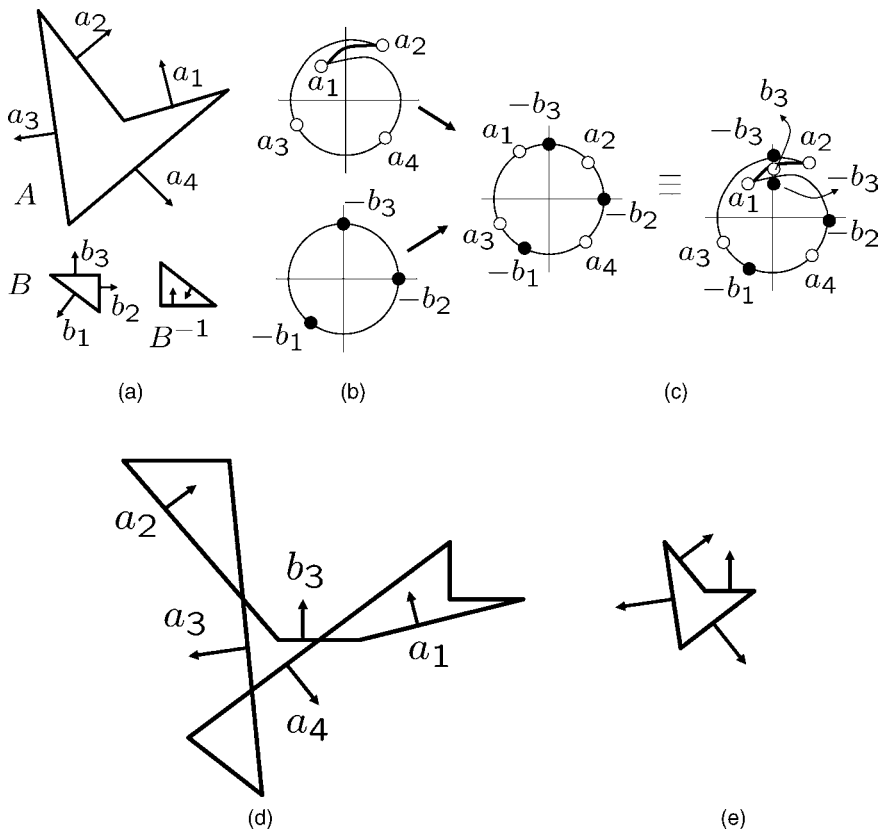


Figure 6.5 The Minkowski decomposition of nonconvex polygons by means of slope diagrams: (a) operand polygons A and B^{-1} ; (b) the slope diagram representations; (c) the merged slope diagram; (d) the boundary addition $\partial A \uplus \partial B^{-1}$; (e) $\partial(A \ominus B) = \text{Pos}(\partial A \uplus \partial B^{-1})$

6.2.2 Boundary Operations on Nonconvex Polygons – More Complex Cases

We have shown the boundary addition of simple nonconvex polygons by means of slope diagrams. It may be pointed out that addition with a nonconvex polygon, even if it is a simple polygon, may result in a complex multiply connected polygon, as shown in Figure 6.6. Of course, even though this will be more involved, the boundary addition process does not need any modification. In this subsection, we will present some cases in which the operation requires a little more care, especially in the merging process.

Even when both of the summands are simple polygons, the basic notion of boundary addition remains exactly the same as before. An example of such an addition is shown in Figure 6.7.

However, cases may arise in which the actual execution of the addition process may appear to be more intricate. Consider, for example, the addition of A and B^{-1} as shown in Figure 6.8(a).

The traversal from \vec{a}_1, \vec{b}_1 to \vec{a}_2, \vec{b}_2 along the slope diagram is straightforward (Figure 6.8(b)). The difficulty arises in the next portion of the traversal. If we try to go to \vec{a}_3 first, we encounter

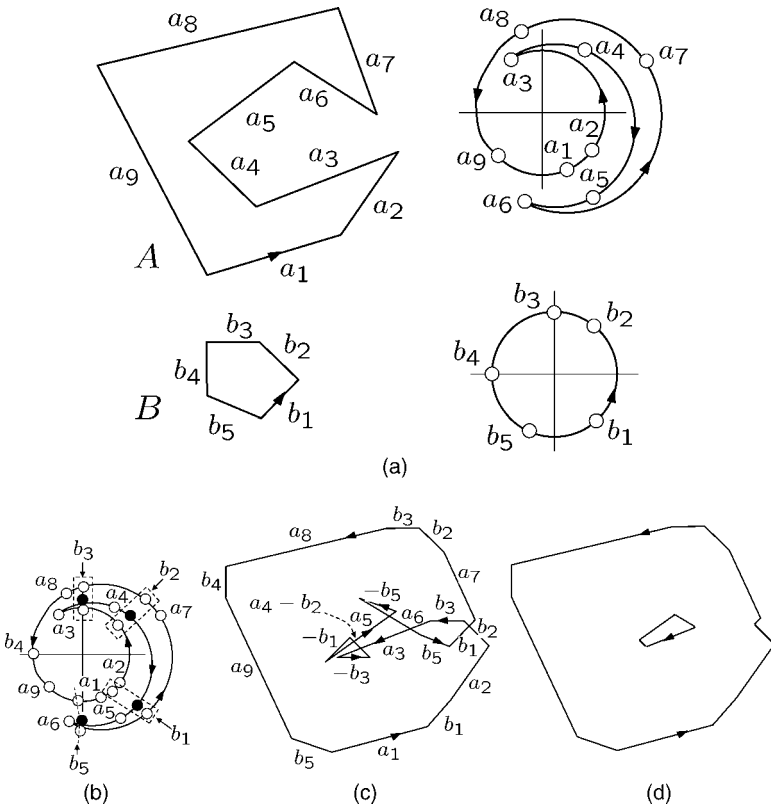


Figure 6.6 A complex example of addition with a simple polygon: polygons A and B and their slope diagrams; (b) the merged slope diagram; (c) the boundary sum $A \cup B$; (d) the Minkowski addition $A \oplus B = \text{Pos}(A \cup B)$

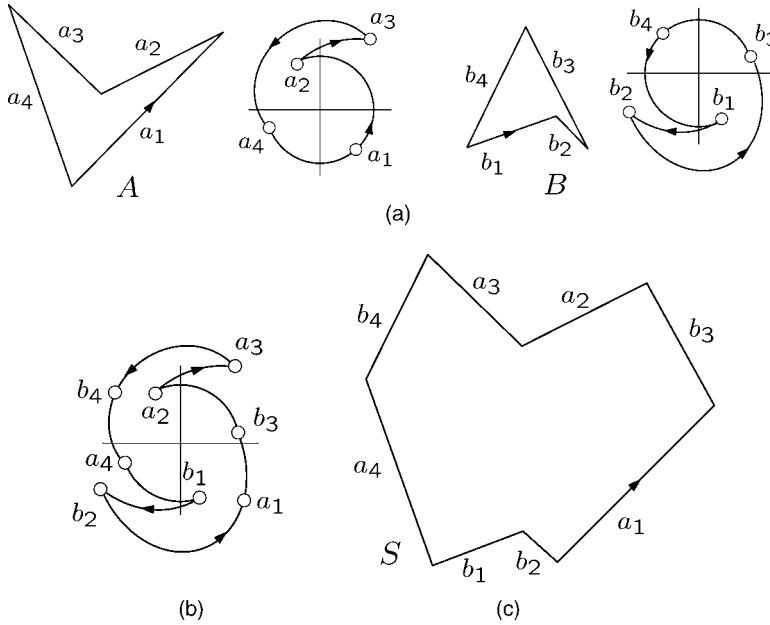


Figure 6.7 The boundary addition of two simple polygons: (a) polygons A and B and their slope diagrams; (b) the merged slope diagram; (c) $S = A \oplus B = \text{Pos}(A \cup B)$

\vec{b}_4 , which directs us to go first to \vec{b}_3 ; on the other hand, if we try to go to \vec{b}_3 first, we encounter \vec{a}_4 , which directs us to go first to \vec{a}_3 . In such a situation, the appropriate technique is to go to \vec{a}_3 and \vec{b}_3 *simultaneously*, as shown in Figure 6.8(c)(iii).

We can reason out the technique in the following way. First, note that the problem is localized. We have to pay attention to one portion of the merged slope diagram; that is, the portion from \vec{b}_2 to \vec{b}_5 (in a counterclockwise sense) in the case of our example figure (Figure 6.8(b)). The rest of the slope diagram (that is, from \vec{b}_5 to \vec{b}_2 in a counterclockwise sense) can be processed in the usual manner. Second, we can view such a situation as a simultaneous occurrence of two events; that is, the slope diagram of B is embedded into that of A , and, simultaneously the slope diagram of A is embedded into that of B . Now to represent, say, the former event, we must ensure that the following conditions hold true (see Figure 6.8(b)): the edge point \vec{b}_3 lies between \vec{a}_2 and \vec{a}_3 , between \vec{a}_3 and \vec{a}_4 (clockwise sense), and also between \vec{a}_4 and \vec{a}_5 ; the edge point \vec{b}_4 lies between \vec{a}_2 and \vec{a}_3 . The portion of the slope diagram of A between \vec{a}_2 and \vec{a}_3 poses no problem, since both \vec{b}_3 and \vec{b}_4 are present in that portion and could be inserted straightaway. But in the next portion – that is, between \vec{a}_3 and \vec{a}_4 – we cannot insert \vec{b}_3 alone, since we have already traversed \vec{b}_4 in the previous portion. Therefore, even in this portion, both \vec{b}_3 and \vec{b}_4 need to be inserted. For a similar reason, in the portion between \vec{a}_4 and \vec{a}_5 , not only \vec{b}_3 but also \vec{b}_4 is inserted. The complete event is depicted by means of the slope diagram shown in Figure 6.8(c)(i). Next, in order to represent the latter event – that is, to embed the slope diagram of A into that of B – we can use an exactly similar set of arguments and show that it will be represented by the slope diagram shown in Figure 6.8(c)(ii). Since both of these

events happen simultaneously, the entire situation is expressed by merging the two slope diagrams as shown in Figure 6.8(c)(iii). No further reduction in the slope diagram is possible that will satisfy all the conditions. The corresponding boundary addition $A \uplus B^{-1}$, and the positive region $\text{Pos}(A \uplus B^{-1})$, which equals $A \ominus B$, are depicted in Figures 6.8(d) and (e), respectively.

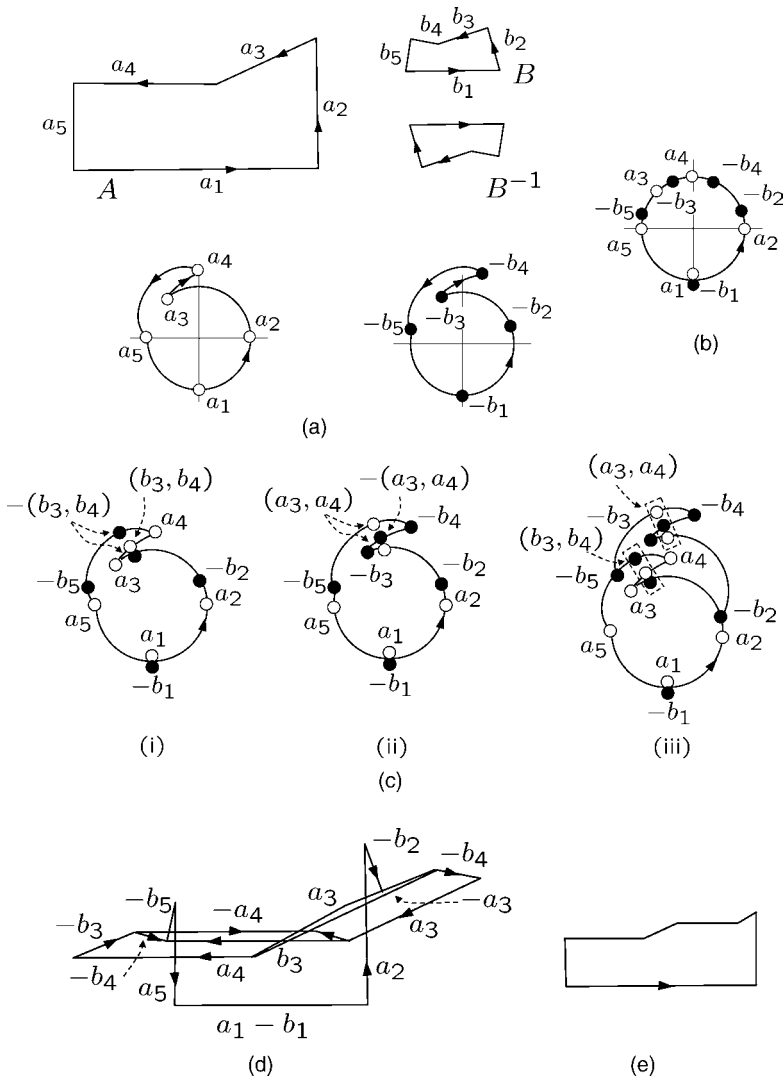


Figure 6.8 A complex example of the boundary addition of two simple polygons: (a) polygons A and B^{-1} and their slope diagrams; (b) the merged slope diagram; (c) interpretation of the merged slope diagram; (d) the boundary sum $A \uplus B^{-1}$; the Minkowski decomposition $A \ominus B = \text{Pos}(A \uplus B^{-1})$; (f) the sum polygon obtained by keeping b_4 dormant; (g) the sum polygon obtained by keeping b_3 dormant; (h) the Minkowski decomposition $A \ominus B = (A \ominus B_1) \cap (A \ominus B_2)$

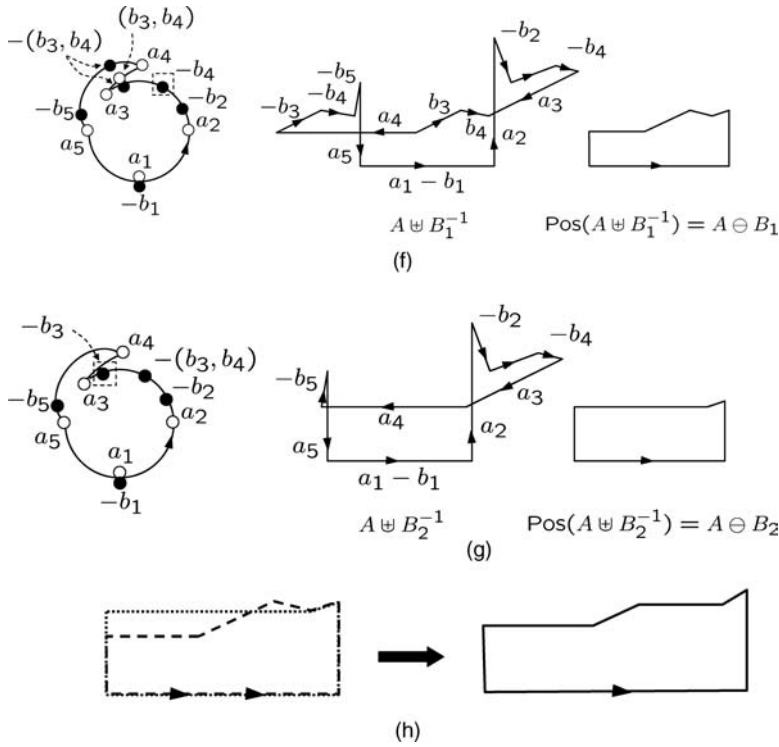


Figure 6.8 (Continued)

Interestingly, the above technique suggests an alternative (though closely related, but relatively simpler) method of determining the Minkowski sum or decomposition (that is, $\text{Pos}(A \uplus B)$ or $\text{Pos}(A \uplus B^{-1})$) directly. We can sketch the basic idea by means of the same example. Assume, where the traversal path is complicated, that only one edge point of B^{-1} is effective at a time while others are dormant, and consider each edge point as being effective in turn. For example, in traversing from \vec{a}_2 to \vec{a}_3 in the slope diagram, we can first assume that \vec{b}_4 is dormant and \vec{b}_3 is effective, while in the second turn we assume \vec{b}_3 to be dormant and \vec{b}_4 to be effective. We have to take the *union* of the positive regions of all these sum polygons thus obtained if both the summands are of the same type and the *intersection* if they are of opposite types. We clarify this technique by means of Figures 6.8(f)–(h).

The correctness of this method can be understood from set-theoretic considerations. What we are effectively doing is breaking B up into B_1, B_2, \dots , such that each B_i is equal to B minus the corresponding dormant edge points, and the union of all these B_i 's will be equal to B ; that is, $B = B_1 \cup B_2 \cup \dots$. Finally, we make use of the following results:

$$A \oplus (B_1 \cup B_2 \cup \dots) = (A \oplus B_1) \cup (A \oplus B_2) \cup \dots,$$

$$A \ominus (B_1 \cup B_2 \cup \dots) = (A \ominus B_1) \cap (A \ominus B_2) \cap \dots$$

For further details, see [28].

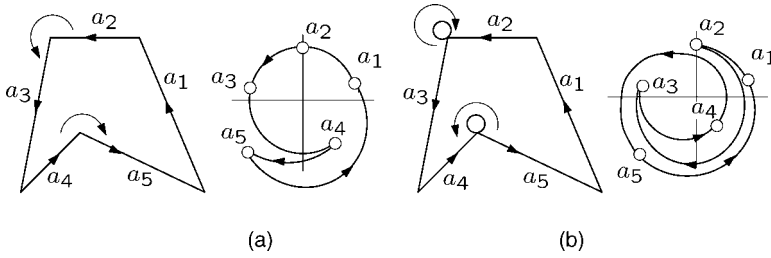


Figure 6.9 The difference between the slope diagrams of a simple polygon and a self-crossing polygon: (a) a simple polygon and its slope diagram; (b) a self-crossing polygon and its slope diagram

Although in the process of boundary addition, self-crossing polygons are being generated, we have decided not to include a general procedure for adding two self-crossing polygons here. It must be clear to you by now that given the way in which we have formulated the boundary addition problem, the addition procedure that we have so far developed cannot be directly applied in the case of general self-crossing polygons without appropriate modifications. In other words, we do not regard a convex or a simple polygon as special cases of a self-crossing polygon. One reason is that while the structure of convex and simple polygons demands that the transversal angle from one edge point to the next must be less than 180° , it may be more than 180° in the case of self-crossing polygons. This is clearly demonstrated in Figure 6.9.

6.2.3 Nonconvex Polyhedra and the Slope Diagrammatic Approach

The slope diagram of a nonconvex polyhedron, like that of a convex one, can be captured on a unit sphere by indicating the facet points, edge arcs, and vertex regions. A typical such example is shown in Figures 6.10(a) and (b). Note that the edge arc connecting the facet points f_2 and f_3 is drawn using heavy black lines to indicate its negative sense. The projected slope diagram is also shown in Figure 6.10(c).

The boundary addition \cup of nonconvex polyhedra also remains exactly identical; that is, merging of the slope diagrams of the operands, and then realization of the boundary sum from the merged slope diagram. In case of nonconvex polyhedra too, like the nonconvex polygonal case, the boundary sum will be a self-crossing polyhedron, in general, even if both the summands are positive polyhedra.

In Figure 6.11, we present an example of Minkowski addition by means of boundary addition and the Pos operation.

6.3 A Unified Algorithm for Minkowski Operations

6.3.1 The Unified Algorithm

We can summarize our discussion by saying that we have arrived at a unified approach to computing Minkowski operations on boundary-represented geometric objects. We find that:

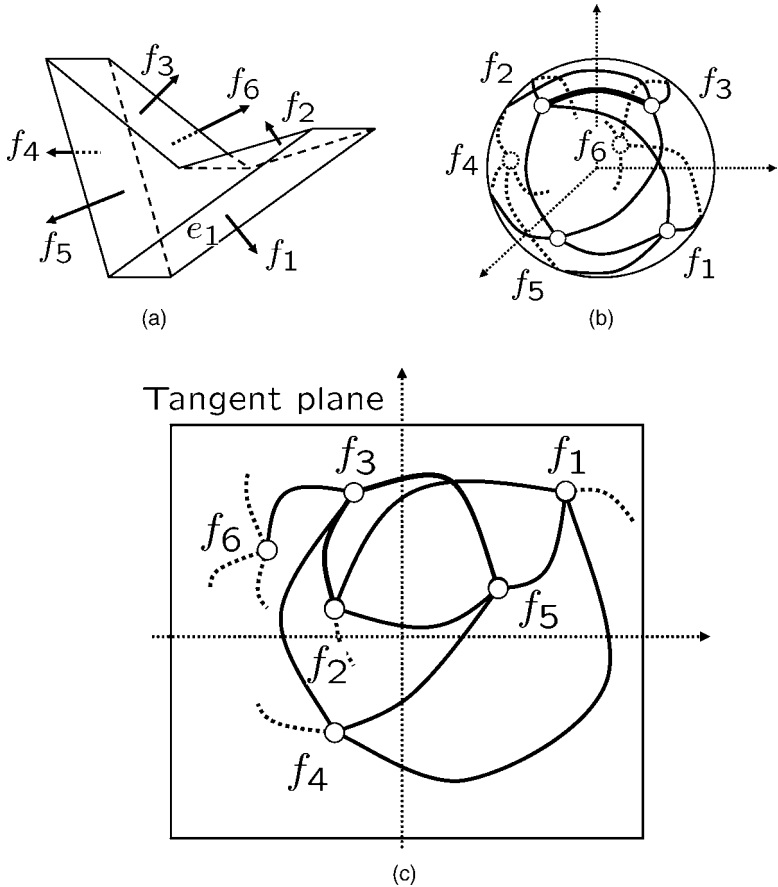


Figure 6.10 A nonconvex polyhedron and its slope diagram representation; the projected slope diagram is also shown (not to scale): (a) a typical nonconvex polyhedron; (b) the slope diagram of the polyhedron on a unit sphere (not all of the edge arcs are shown); (c) the projected slope diagram of the nonconvex polyhedron (the dark arc denotes the nonconvex edge of the polyhedron)

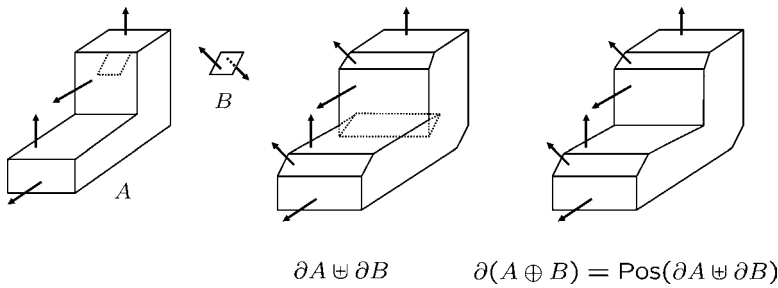


Figure 6.11 Minkowski addition: the boundary addition of two positive polyhedra, and the behavior of the sum at the nonconvex portion

- Both Minkowski addition and decomposition are essentially the same operation; while Minkowski addition is basically a boundary addition operation of two positive objects, Minkowski decomposition is the boundary addition of one positive object and one negative object.
- Minkowski operations on both convex and nonconvex objects are essentially the same; the only extra consideration needed is to treat the nonconvex faces as the faces of a negative object.
- Minkowski operations on both two-dimensional and three-dimensional objects (in fact, general d -dimensional objects) are exactly alike; they eventually reduce to addition/subtraction of real numbers.

Below, we present our *unified algorithm* to compute Minkowski operations on polygonal and polyhedral objects.

6.3.1.1 A Unified Algorithm to Compute Minkowski Operations

(The input operands are polygons or polyhedra that are specified in their boundary-represented forms.)

A. Computation of the boundary addition operation \uplus

1. *The formation of slope diagrams.* Represent the operands in their slope diagram forms. (In the case of computing $A \oplus B$, the operands are ∂A and ∂B , while the operands are ∂A and ∂B^{-1} in computing $A \ominus B$.)
2. *Merging of slope diagrams.* Merge the slope diagrams of the two operands into a single one.
3. *Realization of the boundary sum.* From the merged slope diagram, realize the polygon or polyhedron that it represents. This polygon or polyhedron is the boundary sum $\partial A \uplus \partial B$ or $\partial A \uplus \partial B^{-1}$, which is, in general, a self-crossing geometric object. Let us call it S_{boun} .

B. Computation of Minkowski operations

4. *Determination of $\text{Pos}(S_{\text{boun}})$.* Compute the boundary of the positive portion of S_{boun} . This will be equal to $\partial(A \oplus B)$ or $\partial(A \ominus B)$.

6.3.2 A Complexity Analysis of the Unified Algorithm

Consider the complexity of the algorithm for two-dimensional polygons. Let n_1 and n_2 be the number of edges of the input polygons. Step 1 of the algorithm can then be carried out in $O(n_1 + n_2)$; that is, in $O(n)$ time, where $n = n_1 + n_2 = \text{total size of the input}$. Let k be the number of edges of the boundary sum S_{boun} . Steps 2 and 3 then require $O(k)$ time. Thus the computation of S_{boun} takes $O(n + k)$ time in total. Note that the size of k may vary from $n_1 + n_2$ (in case of Minkowski addition of two convex polygons) to $O(n_1 n_2)$ (in the most general case). In computing $\text{Pos}(S_{\text{boun}})$ – that is, in Step 4 – the basic algorithmic step is to “determine all the intersections among k given straight-line segments in the plane.” This can be done by using the method of Bentley and Ottman [82], that time complexity of which is $O((k + m) \log k)$, where m denotes the number of intersections among the k line segments. There is another improved, but slightly difficult, algorithm proposed by Chazelle [82] the time

complexity of which is $O(m + (k \log^2 k / \log \log k))$. Thus the overall time complexity of the unified algorithm will be $O(n + k + m + (k \log^2 k / \log \log k))$. Note that, in general,

$$m < \binom{k}{2} = O(k^2).$$

In the case of three-dimensional polyhedra, let n_1 and n_2 be the numbers of edges of the two input polyhedra. Since the numbers of vertices, edges, and facets of a polyhedron without any hole follow Euler's formula,

$$(\text{number of vertices}) - (\text{number of edges}) + (\text{number of facets}) = 2,$$

we can say that the numbers of vertices and facets of a polyhedron are related to its number of edges by a small constant factor. In other words, we say that the sizes of the input polyhedra are of the orders $O(n_1)$ and $O(n_2)$, respectively. Therefore, Step 1 of the unified algorithm can be carried out in linear time – that is, in $O(n)$ time – where $n = n_1 + n_2$. If k denotes the total number of edges in S_{boun} , then Steps 2 and 3 take $O(k)$ time. Therefore, the computation of S_{boun} takes $O(n + k)$ time in total. The major computational cost in the three-dimensional case arises in Step 4. The crudest approach to the computation of $\text{Pos}(S_{\text{boun}})$ consists of determining the intersections of each facet of S_{boun} with every other facet. It is easy to verify that such an approach can take as much as $O(k^2)$ time. Clearly, this step dominates the initial step of determining S_{boun} , and the overall time complexity of the algorithm becomes $O(k^2)$ – that is, $O(n^4)$ – in the worst case.

6.3.3 Simplification of the Unified Algorithm Depending on the Type of Input

At this point, we must clearly state that the unified algorithm presented above is a general framework in which the underlying methodology of computation of Minkowski operations is expressed in an algorithmic form; but we do not stress that exactly the same algorithm should be used in every case. In other words, if the input set is endowed with more structures – for example, if both the operands are convex, and so on – it is necessary to modify the algorithm and the data structures appropriately in order to increase the computational efficiency. Here, we consider a few special cases to demonstrate how the structure of the input set can be exploited fully to make the algorithm more efficient. We also show how, in certain cases, the unified algorithm automatically reduces to a simpler algorithm, since some of the computational steps may no longer be required.

6.3.3.1 Two-Dimensional Cases

1. *Minkowski addition of two convex polygons.* In Section 5.5.3, we have given an algorithm for addition when both of the input polygons A and B are convex. Step 4 of the unified algorithm is not required, since $\partial(A \oplus B) = \text{Pos}(\partial A \uplus \partial B)$ in this case. Therefore, the complexity of the algorithm becomes $O(n + k)$. Since, in this case, $k = n_1 + n_2 = n$, the complexity is linear; that is, $O(n)$. Interested readers can also refer to an algorithm by Schwartz [89].
2. *Minkowski decomposition of two convex polygons.* In Section 5.5.3, we have also discussed how to determine $A \ominus B$, when both A and B are convex. The same algorithm is discussed

in detail in Ghosh [28,30]. It is easy to see that the computation of the boundary sum $\partial A \uplus \partial B^{-1}$ will take $O(n)$ time, since $k = n_1 + n_2 = n$ in this case too. However, Step 4 cannot be avoided, since the boundary sum is, in general, a self-crossing polygon. But it is quite easy to obtain a much faster algorithm to execute Step 4, since the n line segments in the boundary sum are not arbitrary line segments in the plane; these segments are, in fact, translated edges of the convex polygons A and B . In [28], an $O(n_1)$ algorithm for Step 4 is given, where n_1 denotes the number of edges of A . Thus the overall time complexity again reduces to $O(n)$. We also refer to Guibas *et al.* [35].

3. *Minkowski decomposition $A \ominus B$ where A is a convex polygon.* Consider the decomposition $A \ominus B$, where A is a convex polygon, but B is a general polygon, which is not necessarily convex. From the set-theoretic result, we know that if A is a convex set, then

$$A \ominus B = A \ominus C(B),$$

where $C(B)$ denotes the convex hull of B . Therefore, to obtain an efficient algorithm, it is advisable to determine the convex hull of B first, and then to use the *convex-convex decomposition* algorithm as discussed above. Let n_1 and n_2 be the numbers of edges of A and B , respectively. To determine $C(B)$, we can use any standard convex hull construction algorithm, such as Graham's scan algorithm [82], which runs in $O(n_2 \log n_2)$ time. As shown above, $A \ominus C(B)$ takes $O(n)$ time, where n denotes the total size of the input polygons. Therefore, $A \ominus B$ can be determined in $O(n + n_2 \log n_2)$ time.

4. *Minkowski operations on two planar regions whose boundaries are smooth curves.* First, consider the addition $A \oplus B$. We assume that the boundary curves of both A and B can be represented by smooth analytic functions. Let us denote them, in the parametric form, as follows:

$$\partial A = [A_x(t_1), A_y(t_1)], \quad (6.4)$$

$$\partial B = [B_x(t_2), B_y(t_2)], \quad (6.5)$$

where t_1, t_2 are two scalar quantities and are in closed intervals on the t_1 -axis and the t_2 -axis, respectively; $A_x(t_1)$ represents a polynomial function of t_1 , and so on. We assume that ∂A and ∂B are oriented in the sense corresponding to an increase in the parameters t_1 and t_2 , respectively.

The corresponding faces $F(A, u)$ and $F(B, u)$, in this case, mean the points of ∂A and ∂B , respectively, where the tangent lines are parallel and similarly oriented (Figure 6.12). Therefore, the determination of $\partial A \uplus \partial B$ reduces to the following sequence of operations: (a) For every point $a \in \partial A$, find the direction of the tangent line, and then determine the corresponding point – say, $b \in \partial B$ – where the tangent line is parallel and similarly oriented; then (b) vectorially add a and b – that is, $a + b$. This method is clearly depicted in Figure 6.12.

In certain circumstances, it may be possible to determine $\partial A \uplus \partial B$ completely analytically. Let $a = [A_x(t'_1), A_y(t'_1)]$, for some $t_1 = t'_1$. To obtain the corresponding point $b \in \partial B$, we have to solve the following equation for t_2 :

$$\left[\frac{\delta B_y(t_2)/\delta t_2}{\delta B_x(t_2)/\delta t_2} \right] = \left[\frac{\delta A_y(t_1)/\delta t_1}{\delta A_x(t_1)/\delta t_1} \right]_{t_1=t'_1}. \quad (6.6)$$

Here $\delta X(t)/\delta t$ denotes the differentiation of the function $X(t)$ with respect to t .

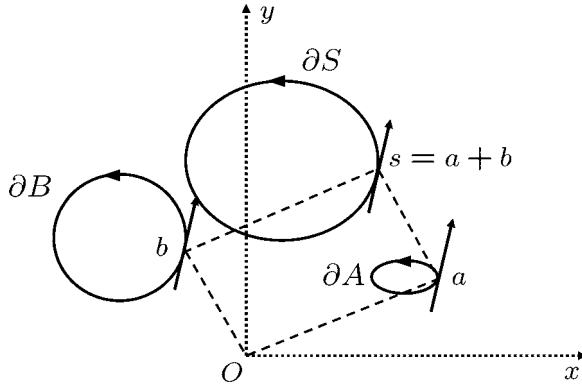


Figure 6.12 The Minkowski addition of two regions bounded by smooth curves

Equation (6.6) yields the solution for t_2 in terms of t'_1 . Let $t_2 = h(t'_1)$, where $h(t'_1)$ denotes some polynomial in t'_1 . Then the corresponding point b can be expressed as $b = [B_x(h(t'_1)), B_y(h(t'_1))]$. Therefore, the analytical expression for the boundary sum will be

$$\partial A \uplus \partial B = [A_x(t_1) + B_x(h(t_1)), A_y(t_1) + B_y(h(t_1))]. \quad (6.7)$$

Whether or not $\partial A \uplus \partial B$ will be a self-crossing curve clearly depends on the nature of ∂A and ∂B . Therefore it is not possible, in general, to comment on the overall complexity of the algorithm, which involves determination of $\text{Pos}(\partial A \uplus \partial B)$ if the boundary sum is self-crossing. However, in special cases where both A and B are convex, we may get constant time algorithms.

For further details, see Ghosh [25,28]. See also Horn and Weldon [42], where there is a mention of sum of two *extended circular images*. An extended circular image of a shape is somewhat analogous to the slope diagram representation of the shape.

Recently, Sugihara *et al.* [95,52] have shown that the Minkowski sum and its inverse operation can be defined very easily for nonconvex figures as well by slightly changing the figure representation with parallel axes, each defined as a rotational sweep of a slope-monotone closed curve. They consider figures that could be represented by the union of a finite number of convex figures. Representation is performed by means of closed curves that exhibit monotonic change of the tangent direction and that allow self-intersections. These closed curves consist of “visible” portions (the figure boundaries) and “invisible” portions (the figure interior).

Just as described in this book, the use of such a concept to explicitly represent “invisible” portions proves very effective. In fact, figure manipulation becomes very simple; besides, if only results obtained for “visible” portions are taken into consideration, this corresponds to the conventional concept of the Minkowski operations. In this sense, the new approach advanced in their study is a kind of generalization of the Minkowski operations, one that offers a simple implementation of the scheme that we have set out in this chapter.

6.3.3.2 Three-Dimensional Cases

5. *Minkowski addition of two convex polyhedra.* From the unified algorithm, it is not difficult to derive the following result.

Theorem 6.1. *Let A and B be two convex polyhedra. Let $v_i^A (i = 1, \dots, n_1)$ be the (position vectors of the) vertices of A , and let $v_j^B (j = 1, \dots, n_2)$ be the vertices of B . Then the sum*

$$A \oplus B = C \left(\left\{ v_i^A + v_j^B \mid i = 1, \dots, n_1, j = 1, \dots, n_2 \right\} \right),$$

where $C(X)$ represents the convex hull of a set X .

This result can be used to devise a very simple two-step algorithm to add one convex polyhedron to another convex polyhedron. The first step is to vectorially add every vertex of A to every vertex of B . The total number of points thus generated will be $n_1 n_2$, so this step takes $O(n_1 n_2)$ time. The second step is to determine the convex hull of these $n_1 n_2$ points in the three-dimensional space. Using some standard algorithm, say the Preparata–Hong algorithm [81], this can be accomplished in $O(n_1 n_2 \log n_1 n_2)$ time, which clearly dominates the computation of the first step.

For the addition of convex polyhedra, see also Guibas and Seidel [36].

6. *Minkowski addition of a space curve by a spherical ball.* We have already shown that if the boundaries of two-dimensional operands can be expressed as smooth analytic functions, it may be possible to obtain the boundary of the product object purely analytically. Here, we present an example to demonstrate this for three-dimensional operands as well.

Let A be a space curve, whose parametric equation is given by

$$\partial A = [A_x(t), A_y(t), 0],$$

and let B be a sphere of radius r , whose boundary equation is

$$\partial B = [x, y, z], \quad \text{where } x^2 + y^2 + z^2 = r^2.$$

The center of the sphere is assumed to be at $(0, 0, 0)$.

Let a be a point on the space curve – say, $a = \partial A(t = t')$. We have to find the corresponding point(s) $F(B, u)$ on ∂B . Since A is a space curve, the outer normals at a form a plane, called the *normal plane* at that point (Figure 6.13(b)). The equation of the normal plane can be obtained in the following way.

The unit tangent vector w of ∂A at a is given by

$$w = \left[\frac{(\delta A / \delta t)}{|(\delta A / \delta t)|} \right]_{t=t'} = \left[\frac{\dot{A}_x(t)}{l}, \frac{\dot{A}_y(t)}{l}, 0 \right]_{t=t'},$$

where $\dot{A}_x(t) = \delta A_x(t) / \delta t$, and so on, and $l = \sqrt{(\dot{A}_x(t))^2 + (\dot{A}_y(t))^2}$. (Hereafter, we shall write A_x , and so on, instead of $A_x(t)$.)

One point of the normal plane is the point a , and the unit tangent vector w is normal to that plane. Therefore, the equation of the normal plane at a is given by

$$\frac{\dot{A}_x}{l}(x - A_x) + \frac{\dot{A}_y}{l}(y - A_y) + 0 \cdot (z - 0) = 0,$$

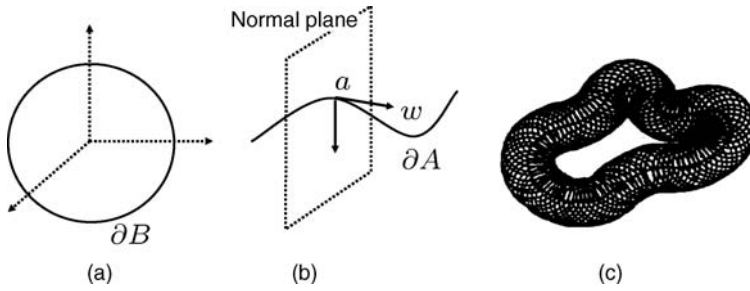


Figure 6.13 The Minkowski addition of a space curve by a spherical ball: (a) a sphere of radius r ; (b) a space curve and its normal plane at the point a ; (c) a typical S_{boun} for Minkowski addition of a closed space curve and a sphere

or

$$\dot{A}_x(x - A_x) + \dot{A}_y(y - A_y) = 0,$$

where $t = t'$.

The corresponding points $F(B, u)$ will be the intersection points of ∂B with a plane that is parallel to the normal plane and that passes through the center of the sphere.

The equation of the plane parallel to the normal plane and that passes through the point $(0, 0, 0)$ is

$$\dot{A}_x \cdot x + \dot{A}_y \cdot y = 0,$$

where $t = t'$. Substituting the value of y (in terms of x) from the above equation into the equation for the sphere, we obtain

$$\frac{x^2}{\left(\sqrt{\frac{r^2}{1 + \dot{A}_x^2/\dot{A}_y^2}}\right)^2} + \frac{z^2}{r^2} = 1.$$

Similarly, substituting the value of x , we obtain

$$\frac{y^2}{\left(\sqrt{\frac{r^2}{1 + \dot{A}_y^2/\dot{A}_x^2}}\right)^2} + \frac{z^2}{r^2} = 1.$$

This means that the intersection points – that is, $F(B, u)$ – can be expressed as

$$x = F_x \cos \theta, \quad y = F_y \cos \theta, \quad z = r \sin \theta,$$

where θ varies from 0 to 2π radians, and

$$F_x = \left(\sqrt{\frac{r^2}{(1 + \dot{A}_x^2/\dot{A}_y^2)}} \right) = F_x(t = t'),$$

$$F_y = \left(\sqrt{\frac{r^2}{(1 + \dot{A}_y^2/\dot{A}_x^2)}} \right) = F_y(t = t').$$

Therefore,

$$x(t, \theta) = F_x(t) \cos \theta + A_x(t),$$

$$y(t, \theta) = F_y(t) \cos \theta + A_y(t),$$

$$z(t, \theta) = r \sin \theta.$$

A typical $\partial A \cup \partial B$ is shown in Figure 6.13(c).

7

The Morphological Decomposability and Indecomposability of Binary Shapes

7.1 The Morphological Indecomposability Problem

7.1.1 The Problem and its Motivation

The problem that we are given here is as follows:

Given a set of points S in the plane, determine whether it can be expressed as a Minkowski sum of two simpler sets of points. In other words, are there sets of points A and B in the plane, such that a given S can be expressed as $S = A \oplus B$?

A and B are two arbitrary sets of points in two-dimensional Euclidean space, and the Minkowski addition operation \oplus again means that

$$A \oplus B = \{a + b \mid a \in A, \text{ and } b \in B\}, \quad (7.1)$$

where “+” denotes the vector addition of two points. The sets A and B are called the summands of the sum S .

The importance of this problem in shape understanding can be gauged from its analogous problem in number theory: “Given a positive integer number n , are there integers $k, l > 1$ such that $n = k \cdot l$?” This question, as we all know, gave rise to one of the most fundamental concepts of number theory; namely, the concept of *prime numbers*. A positive integer p is called prime if it cannot be expressed as a product of two numbers other than in the most trivial manner; that is, as the product of 1 and the number p itself.

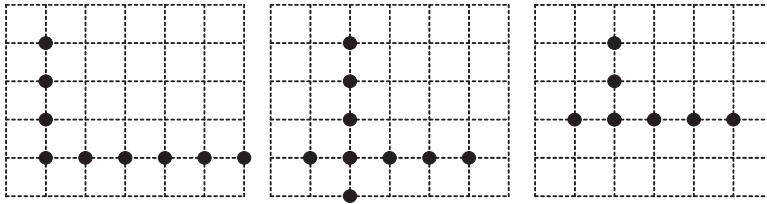


Figure 7.1 A few simple binary shapes that are morphologically indecomposable into further simpler shapes

Analogously, there exist sets of points in the plane or space that cannot be expressed as a Minkowski sum other than in the most trivial manner; that is, as the sum of a single point and the given set S itself. Such sets cannot be decomposed further as a Minkowski sum of two simpler shapes. Such point sets can be called the *morphologically indecomposable shapes*. A few such indecomposable shapes in the domain of binary images are shown in Figure 7.1.

Exactly like the prime numbers, the indecomposable shapes can be considered as the fundamental building blocks of all geometric shapes. (We consider “geometric shapes” as sets of points in real Euclidean space or, formally, as the set of all subsets of E^d .) If we can identify the set of all indecomposable shapes $\{I_1, I_2, \dots\}$, then any point set S can be represented as a Minkowski sum of indecomposable shapes,

$$S = I_i \oplus \dots \oplus I_k. \quad (7.2)$$

The basic hurdle, however, is to identify an indecomposable shape. Just as it is not easy to say immediately whether or not a given integer – for example, 3 263 443 – is a prime number, similarly it is difficult to say offhand whether or not a given point set is indecomposable. Therefore, the basic question is: “Can we find out the necessary and sufficient condition for a point set S to be indecomposable?” From our experience, it appears that it may not be possible to obtain any closed-form mathematical condition for indecomposability. In the case of integer numbers too, you may recall that all attempts to obtain a closed-form formula to recognize a prime number have so far failed. For example, Fermat’s conjecture that numbers of the form $2^{2^n} + 1$ are always prime (a sufficient condition for primality) has been found to be incorrect, as are the Mersenne numbers. To this day, all the basic methods to determine whether or not a given integer n is prime are computation-oriented *trial-and-error* methods. The most popular one, the *trial division* method, is purely computational: divide n by successive primes $p = 2, 3, 5, \dots$ up to \sqrt{n} , until you discover the smallest p that divides n . Almost all of the *primality tests* invented so far are essentially just ingenious ways to speed up the basic trial division process.

This lesson from number theory indicates that one way of determining the indecomposability of a point set S is to select various subsets of S and then to check whether or not any of these subsets is a summand of S . The question of indecomposability then reduces to devising several *indecomposability tests* so that the subset-selection and the summand-checking processes can be considerably speeded up.

There is another useful lesson from number theory. To simplify the primality question, all of the numbers are subdivided into several convenient classes, and the primality question is

studied within a few of them. The class that has been most thoroughly explored is the set of all integer numbers \mathbb{Z} . Another interesting class is the complex Gaussian integers, in which every number has the form $a + ib$, where a and b are integer numbers. The primality question is slightly different in the latter case, since there are several *unitary divisors* within that class; namely, ± 1 and $\pm i$. Note, for example, that numbers such as 2 or 5 are not primes in that domain, since $2 = (1 + i)(1 - i)$ and $5 = (2 + i)(2 - i)$. Similarly, the indecomposability problem can be studied within a more restricted domain of geometric shapes, instead of considering the set of all geometric shapes at one time. For example, this problem has already been studied in some detail in the domain of convex shapes. In this chapter, we investigate the indecomposability problem in the domain of binary images; that is, black and white discrete images in the plane.

Informally, by a “binary image” we mean a two-dimensional array of pixels on a regular square lattice. The pixels can take only two values: one and zero. Since a binary image can be embedded in the Euclidean plane E^2 , we can model it mathematically as a set of points p_{ij} in the plane whose Cartesian coordinates (i, j) are restricted to only integer values.

Before proceeding further, let us mention a couple of points regarding our approach:

- Our exposition will be guided to some extent by our belief that there is a close connection between the primality problem in number theory and the indecomposability problem in geometry. For example, any singleton point set $\{p\}$ behaves like the number 1, since it is trivially a summand of any point set S ; we can always write $S = S_{-p} \oplus \{p\}$, where S_{-p} denotes the translate of the set S by the vector $-p$; that is, $S_{-p} = S \oplus -p$. In the case where S is a compact convex set in E^d , there is another trivial way of expressing it as a Minkowski sum. If λ denotes any real number greater than zero but less than one, then $\lambda S = \{\lambda s \mid s \in S\}$ is a trivial summand of S , for $S = \lambda S \oplus (1 - \lambda)S$. Obviously, such trivial decompositions are not considered as proper decompositions, since as far as “shape” is concerned, the complexities of the components S_{-p} , or λS (in the case of convex sets), do not reduce from that of the given set S . Therefore, just like the definition of primes, the indecomposable point sets are defined to be those sets that cannot be expressed as a Minkowski sum in any nontrivial manner.
- The indecomposability problem is concerned with the intrinsic shape of a geometric object; the position of the object in space is of no relevance in this context. Therefore, we implicitly assume that all the translates of a given point set are equivalent.

7.1.2 Earlier Works

The indecomposability problem in the domain of convex polytopes in E^d has been studied by a number of mathematicians. (A convex polytope, the analogue of a convex polygon in E^2 and a convex polyhedron in E^3 , is a bounded set that can be written as the intersection of a finite number of half-spaces in E^d .) It has long been known that in the domain of convex polygons in E^2 , triangles (and line segments, which are simply degenerate triangles) are the only indecomposable sets. The set of all triangles and line segments is called the *universal approximating class* for convex regions in E^2 , since every convex region in the plane can be approximated arbitrarily closely by a Minkowski sum of triangles and line segments. On the

other hand, for general convex polytopes in E^d , where $d \geq 3$, any such simple, closed universal approximating classes do not exist.

The characterization of indecomposable polytopes in higher dimensions is a difficult problem. Shephard [93] found a sufficient (but not necessary) condition for a polytope to be indecomposable. Meyer [71] later gave the necessary and sufficient condition for indecomposability of polytopes. Meyer's condition is expressed in terms of the rank of a certain set of linear homogeneous equations that can be formed from the supporting functions of a polytope. Meyer also provided a sharp bound on the number of indecomposable summands needed to express a given polytope as a Minkowski sum. A simpler approach, which yields the same results, was presented by McMullen [77], using a translation-invariant representation of polytopes. Smilansky [94] also proved similar results by introducing the concept of a dual of a polytope.

These approaches, except for Shephard's, are highly *algebraic*, and it is difficult to grasp the geometric sense of indecomposability from such treatments. A comparatively more geometric approach was taken by Kallay [45]. In [46], Kallay showed that the property of being a decomposable or indecomposable convex polytope is *invariant* under nonsingular permissible projective transformations.

Since that time, the indecomposability problem has been studied from the viewpoint of algorithmic complexity. Iwano and Steiglitz [43] defined the notion of *strong decomposition* of a polygon to be Minkowski decomposition in which no edge of one summand is parallel to an edge of the other summand. They showed that the problem of determining whether a convex polygon is strongly decomposable is NP-complete. Mount and Silverman [76] proved that the problem of determining the decomposition of a convex polygon into a minimum number of indecomposable pieces is also an NP-complete problem.

Compared to the work in the continuous convex domain, very little has so far been done in the discrete domain, even in the simplest domain of binary images. Kanungo and Haralick [47] investigated the problem in a restricted domain of binary images. Their domain consists of convex, 4-connected binary images, whose convex hull has edges at angles multiples of 45° with respect to the positive x -axis. In this domain, they showed that 13 indecomposable images exist. Independently, Xu [98] also reached a similar result.

It should be explicitly pointed out that we could not find any literature on indecomposability of *nonconvex* objects – neither in any continuous nor in any discrete domain. On the other hand, almost all practical applications of morphology are concerned with images that are highly nonconvex in nature.

In this chapter, we will raise these issues and attempt to provide some solutions.

To deal with the binary images, the usual practice is to choose the discrete domain as the underlying domain. We, on the other hand, transform a binary image into a polygon in the plane and then study its indecomposability. There are two compelling reasons for this. First, some of the results that are already available for continuous convex objects can be used immediately. Second, in the continuous domain the slope diagram technique has been introduced, which deals with Minkowski addition and decomposition of objects that are nonconvex. By transforming a discrete image into a continuous polygon – convex or nonconvex – the slope diagram technique can be employed. In fact, we shall show that in terms of the slope diagram technique, Minkowski addition and decomposition of binary images appear very much like addition and subtraction of *hypercomplex numbers*, and thereby the resemblance between geometric objects and numbers can be more clearly observed.

7.2 A Special Class of Binary Shapes: The Weakly Taxicab Convex (WTC) Polygons

7.2.1 Transforming Binary Images into Polygons

Any binary image M , as already defined, is a set of points on a regular square lattice in the plane. We transform M into a polygon A by taking the *4-connected polygonal cover* of M . Figure 7.2 shows two typical binary images, their 4-connected polygonal covers, and the oriented boundaries of the covers.

Intuitively, the 4-connected polygonal cover A of M is formed by connecting the adjacent *border points* of M by means of 4-connected line segments; that is, line segments that are either horizontal or vertical. (The border points of M make up the set of points of M that are adjacent to the complement of M [51].) The 4-connected polygonal cover may be conceived in various other ways too. One interesting way is to conceive it morphologically; that is, $\text{Polygonal_Cover}(M) = (M \oplus Q) \ominus Q$, where Q denotes a unit square region. In this chapter, however, we are not concerned with how to compute the polygonal cover A of an image. We assume that the 4-connected polygonal cover A is a complete representation – that is, a mathematically unambiguous representation of a binary image M – and in this chapter we deal with binary images through this representation. One of the consequences of this assumption is that the length of every edge of a polygonal cover can be considered to be an integer number.

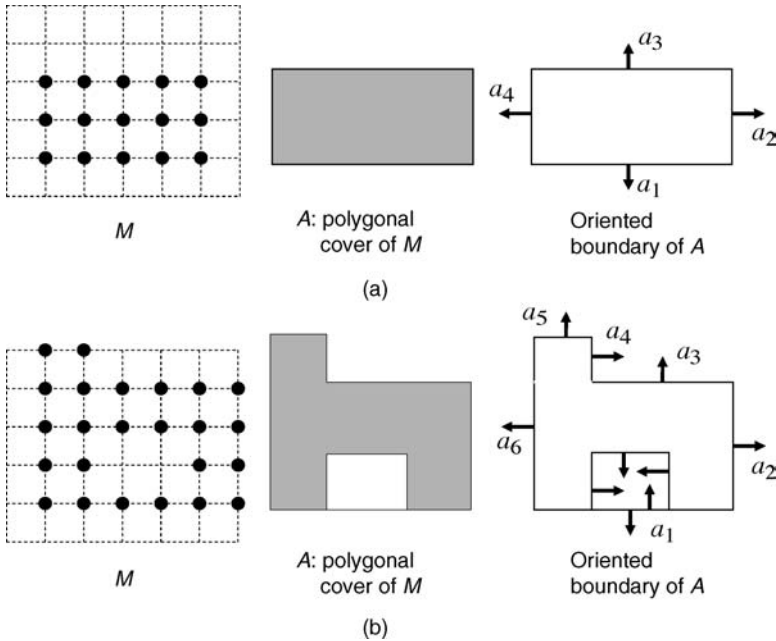


Figure 7.2 Binary images, their 4-connected polygonal covers, and the oriented boundaries of the covers

7.2.2 The Weakly Taxicab Convex Class of Polygons

Our next step is to choose a convenient subset of the set of all binary images. On the question of indecomposability, it is preferable to work with images that are “nearly convex.” The rationale for this conviction comes from the following.

Proposition 7.1. *Let S be a convex set for which there do not exist convex sets A and B such that $S = A \oplus B$. Then there cannot exist general sets P and Q such that $S = P \oplus Q$.*

Proof: Let us assume that there exist general sets P and Q – both of which are not convex – such that $S = P \oplus Q$. Then $C(S) = C(P) \oplus C(Q)$, where “ $C(X)$ ” denotes the convex hull of the set X (see [90], p. 98). Since S is a convex set, $C(S) = S$. This means that there exist convex sets $C(P)$ and $C(Q)$ whose Minkowski sum is equal to S . But this contradicts the given condition. \square

The implications of Proposition 1 are enormous. It implies that if we work within the convex shape domain and discover that some shape S is indecomposable within that domain, then S is intrinsically indecomposable. That is, even in the general shape domain, which consists of all convex as well as nonconvex shapes, S will remain indecomposable.

However, within the domain of 4-connected polygonal covers of binary images, the convex subdomain is not challenging. This subdomain consists of only rectangles, squares, and line segments, which are always decomposable by 2-point binary images; that is, by horizontal and vertical lines of unit lengths (Figure 7.3).

Fortunately, we are able to identify an interesting subdomain consisting of images that are *monotone* with respect to both the x -axis and the y -axis. (A simple polygon is said to be monotone if its boundary is the union of two *edge chains* that are monotone with respect to the same straight line l . An edge chain $C = (v_1, \dots, v_n)$ connecting the vertexes v_1, \dots, v_n by line segments is said to be monotone with respect to l if the orthogonal projections $l(v_1), \dots, l(v_n)$ of the vertexes of C on l are ordered as $\{l(v_1), \dots, l(v_n)\}$. This means that the order of the orthogonal projections and the order of the vertices in the chain are the same. For more details, see Preparata and Shamos [82].)

We can provide an intuitive notion of this image subdomain by means of the examples shown in Figure 7.4. The first and second polygons are monotone with respect to both the x -axis and the y -axis. Unlike the first polygon, the second one is not a convex polygon. But even then, with respect to the x -axis, we may consider that its two edge chains (the lower half and the upper half of the edges) can order the orthogonal projections in a similar way. The same can be done with respect to the y -axis too.

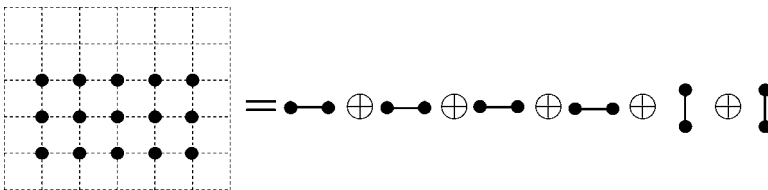


Figure 7.3 The decomposition of a convex 4-connected polygonal cover by 2-point images

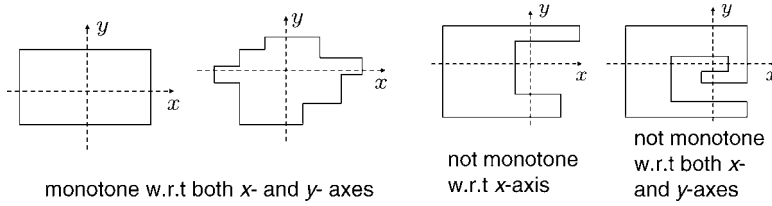


Figure 7.4 Distinguishing “weakly taxicab convex” polygonal covers from other polygonal covers

On the contrary, such monotonic orderings are not possible for the two polygons shown on the right. The first of these two polygons is monotone with respect to the y -axis, but not with respect to the x -axis. The second one is monotone with respect to neither of the axes.

The interesting point is that a 4-connected polygonal cover that is monotone with respect to both the x -axis and the y -axis can be viewed as some kind of convex polygon. To subscribe to this viewpoint, we need the notions of a *generalized straight-line segment* and *generalized convexity* [16, 39, 73].

Definition 7.1 (the generalized straight-line segment): By a straight-line segment between two points a and b , we mean the set of all points that are “between” a and b . If a point c is between points a and b , then

$$\rho(a, b) = \rho(a, c) + \rho(c, b),$$

where $\rho(a, b)$ denotes a “distance” (metric) function between the two points a and b .

Definition 7.2 (generalized convexity): A set K is called convex if at least one line segment joining each pair of points of K lies entirely in K .

In the greater part of our two-dimensional geometry, we consider that the underlying set is the real number space R^2 , and that the distance function is the Euclidean distance function ρ_E ; that is,

$$\rho_E(a, b) = \left[(x_1 - x_2)^2 + (y_1 - y_2)^2 \right]^{1/2},$$

where the coordinates of the points a and b are (x_1, y_1) and (x_2, y_2) , respectively, and x_1, y_1 , and so on are all real numbers. According to the above definition of ρ_E , it is easy to see that the notion of generalized convexity becomes synonymous with the conventional concept of convexity.

However, if instead of ρ_E we use the *taxicab distance* ρ_T as our distance function, which is defined as

$$\rho_T(a, b) = |x_1 - x_2| + |y_1 - y_2|,$$

then a straight-line segment between two points will appear, in general, like a step edge (Figure 7.5(a)). Such a line segment is called a *taxicab line segment* [56, 70, 73].

One basic difference that ρ_T function introduces is that, in general, there will be infinitely many taxicab line segments connecting two points a and b (Figure 7.5(b)), unlike the ρ_E case, where there is one and only one segment. (In fact, any curve from a to b becomes a taxicab

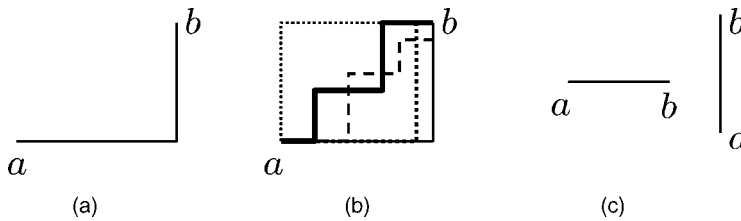


Figure 7.5 Taxicab straight-line segments between two points in R^2 : (a) a taxicab line segment between two points, a and b ; (b) many taxicab segments connecting two points, between two horizontal or vertical points

segment if it varies monotonically from a to b .) Note that only in special situations when a and b lie on a horizontal or a vertical line do the taxicab line segment and the Euclidean line segment appear to be identical (Figure 7.5(c)).

It now follows immediately that in terms of taxicab line segment, our chosen polygonal subdomain is a convex domain. For any polygonal cover A in this subdomain, at least one taxicab line segment joining any two points in A lies entirely within A .

Definition 7.3 (the weakly taxicab convex polygon): A 4-connected polygonal cover A is called a weakly taxicab convex (WTC) polygon if at least one taxicab line segment joining each pair of points of A lies entirely in A .

This means that a 4-connected polygonal cover, which is monotone with respect to both the x -axis and the y -axis, is a weakly taxicab convex polygon. Figure 7.4 shows examples of polygonal covers that are either weakly taxicab convex or not.

The term “weakly” is used to distinguish this kind of polygon from a strongly taxicab convex polygon, where all of the taxicab segments between any two points in the polygon lie entirely within the polygon. Such a strong taxicab convex polygon will be a rectangle, a square, or a horizontal or vertical line. Obviously, any strong taxicab convex polygon is also a WTC polygon. Note that in case of conventional convexity, the distinction between “strongly” and “weakly” convex polygons disappears completely, since only one Euclidean line segment exists between any two points.

Note: The term “generalized convexity” is adopted from Danzer, Grünbaum, and Klee [16], although the notion is akin to the concept of M -convexity. (For a succinct introduction to M -convexity, see Heijmans [39].) There has been considerable terminological uncertainty regarding the distance function ρ_T . In standard books on metric geometry, such as Millman and Parker [73] or Krause [56], ρ_T is called the “taxicab distance,” and the geometry based on the ρ_T -metric is therefore referred to as “taxicab geometry.” We follow this terminology in this book. Terms such as “city block” distance, “Manhattan” distance, or “rectilinear” distance are also frequently used. In the case that the metric ρ_T is supplied by a *norm* (a norm is a generalization of the concept of the length of a vector), the corresponding norm is called the L_1 -norm. In some of the recent literature, instead of “taxicab line segment,” terms such as “staircase path” or “xy-path” are used; the latter term indicates that the path is monotone with respect to both the x and y directions. In referring to the set of polygons whose boundary edges are either horizontal or vertical, we find usages such as “rectilinear,” “iso-oriented,” “isothetic,” “orthogonal,” and so on, and, therefore, a polygon from that set that appears to be convex according to some convexity definition (not necessarily the one we have used) is often referred to as an “iso-convex” or “ortho-convex” polygon. We call such a polygon a “taxicab convex” polygon, indicating that the convexity notion that we use here is based on the taxicab metric. In short, no terminological guidelines concerning these notions have evolved yet, and we have to depend entirely on the definitions given in a particular context.

Let us mention that an interesting undertaking will be to investigate the geometric properties of WTC polygons. For example, we can conjecture the following property:

A set is a weakly taxicab convex set if it is connected, and any vertical or horizontal line intersects the set in a single segment.

The interesting part of this property is that our generalization of the notion of convexity becomes immediately apparent, the idea being to replace the “set of all lines” by the “set of horizontal and vertical lines.” In this chapter, however, we shall not go into such discussions, except to mention a few properties of WTC polygons that are relevant for Minkowski operations.

7.2.3 A Few Properties of WTC Polygons Related to Minkowski Operations

Proposition 7.2. *If A and B are two WTC polygons, their Minkowski sum $A \oplus B$ is also a WTC polygon.*

The proof of this proposition requires the following lemma.

Lemma 7.3. *If L_A and L_B are two taxicab line segments whose end-points are a_1, a_2 and b_1, b_2 , respectively, then there exists at least one taxicab line segment between the points $a_1 + b_1$ and $a_2 + b_2$ that lies entirely within $L_A \oplus L_B$.*

Proof: Without loss of generality, we assume that a_1 and b_1 are at the origin of some chosen coordinate system; that is, $a_1 = (0, 0)$, $b_1 = (0, 0)$. Let $a_2 = (\alpha, \beta)$, $b_2 = (\lambda, \mu)$, where α, λ are the x -coordinates and β, μ are the y -coordinates of the other end-points.

Since L_A, L_B are taxicab segments, they are monotone with respect to both the x - and the y -axes. Without loss of generality, we can assume that both the x -coordinates and the y -coordinates of L_A increase monotonically, the first one from 0 to α and the second one from 0 to β (Figure 7.6(a)). For the other segment L_B , it is sufficient to consider only the following two cases, since any other case will be equivalent to one of these two.

Case I. Both the x -coordinates and the y -coordinates of L_B too, like L_A , increase monotonically, from 0 to λ and from 0 to μ respectively (Figure 7.6(b)). The corresponding $L_A \oplus L_B$ is also shown in the figure.

A taxicab segment from the point $a_1 + b_1$ to the point $a_2 + b_2$ within $L_A \oplus L_B$ can be constructed in the following way: starting from $a_1 + b_1 = (0, 0)$ first go along the line $L_A \oplus \{b_1\}$ to its end-point $a_2 + b_1$, and then from $a_2 + b_1$ go along $L_B \oplus \{a_2\}$ to its end-point $a_2 + b_2$. The taxicab segment thus formed is simply the concatenation of $L_A \oplus \{b_1\}$ and $L_B \oplus \{a_2\}$. It is shown by a heavy line within $L_A \oplus L_B$. (Obviously, we can obtain another taxicab segment by concatenating $L_B \oplus \{a_1\}$ and $L_A \oplus \{b_2\}$.)

Case II. The y -coordinates of L_B , as in the previous case, increase monotonically from 0 to μ . But this time the x -coordinates decrease monotonically from 0 to $-\lambda$ (Figure 7.6(c)). The corresponding $L_A \oplus L_B$ is also shown in the figure.

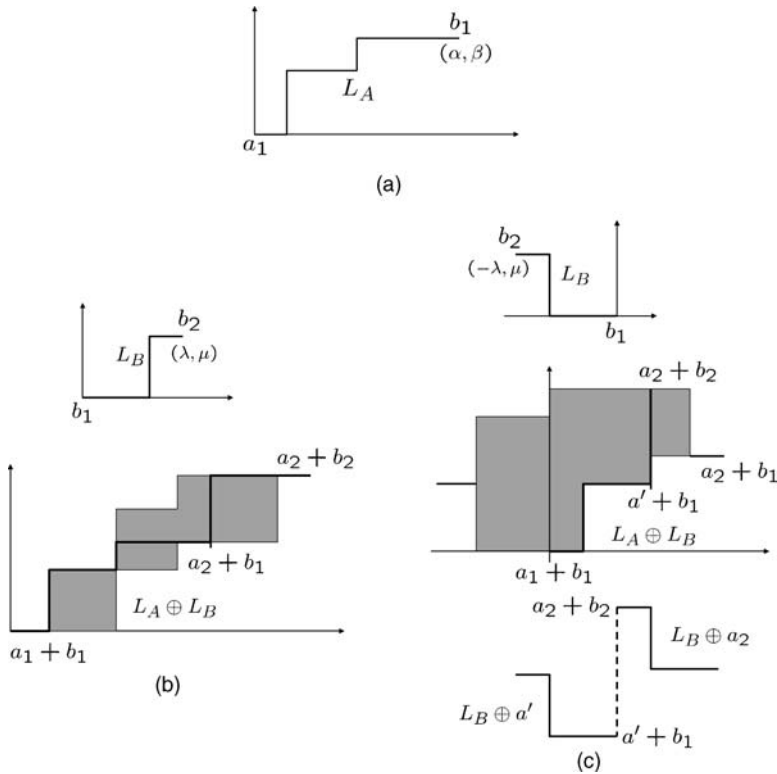


Figure 7.6 The existence of a taxicab segment from $a_1 + b_1$ to $a_2 + b_2$ within the sum of the segments $L_A \oplus L_B$: (a) a typical taxicab line segment L_A ; (b) Case I; (c) Case II

Let us assume that $|\alpha| \geq |\lambda|$. Therefore, the x -coordinate $\alpha - \lambda$ of the point $a_2 + b_2$ will satisfy $0 \leq \alpha - \lambda \leq \alpha$. Identify a point, say a' , on L_A whose x -coordinate is equal to $\alpha - \lambda$.

The construction of a taxicab segment from the point $a_1 + b_1$ to the point $a_2 + b_2$ can proceed in the following way: starting from $a_1 + b_1$, first go along the line $L_A \oplus \{b_1\}$ to the intermediate point $a' + b_1$, and then vertically from $a' + b_1$ to the point $a_2 + b_2$ (shown by a heavy line within $L_A \oplus L_B$).

We now need to prove that the vertical line from $a' + b_1$ to $a_2 + b_2$ lies entirely within $L_A \oplus L_B$. Consider translating the line L_B continuously from a' to a_2 along L_A . The two extremities of those translates – that is, $L_B \oplus \{a'\}$ and $L_B \oplus \{a_2\}$ – are shown separately in Figure 7.6(c). Note that $a' + b_1$ is the rightmost point of the leftmost translate $L_B \oplus \{a'\}$, while $a_2 + b_2$ is the leftmost point of the rightmost translate $L_B \oplus \{a_2\}$. It is obvious that there is no way in which L_B can be continuously translated from a' to a_2 along some taxicab segment without completely covering the vertical line from $a' + b_1$ to $a_2 + b_2$. This means that the vertical line lies entirely within $L_A \oplus L_B$. This completes the proof. \square

Proof of Proposition 7.2: Let us assume that $A \oplus B$ is not a WTC polygon. This means that there are at least a pair of points $p, q \in A \oplus B$, such that no taxicab line segment connecting p and q lies entirely within $A \oplus B$. We can always write $p = a_1 + b_1$ and $q = a_2 + b_2$, where $a_1, a_2 \in A$ and $b_1, b_2 \in B$. Since A is a WTC polygon, there exists at least one taxicab line segment L_A between a_1 and a_2 , which lies entirely within A . Similarly, let L_B be a taxicab line segment connecting b_1 and b_2 , which lies entirely within B . Therefore, $L_A \oplus L_B \subset A \oplus B$. Now, according to Lemma 7.3, it is always possible to draw a taxicab line segment connecting p and q that will lie entirely within $L_A \oplus L_B$, and thereby, within $A \oplus B$. This contradicts our initial assumption. \square

Proposition 7.4. *If A and B are two WTC polygons, then their set intersection $A \cap B$ is either empty or consists of one or more polygons, each of which is a WTC polygon.*

Proof: If A and B are completely disjoint, then $A \cap B = \emptyset$. If that is not the case, it may happen that there are two points $p, q \in A \cap B$, but none of the taxicab lines between p and q in A coincide with any of the taxicab lines between p and q in B (Figure 7.7). In that case, $A \cap B$ gives rise to a number of disjoint polygons C, D, \dots (shown shaded in Figure 7.7). The rest of the proof is to show that any one of these disjoint polygons – say, C – is a WTC polygon. Let us assume that r and s are two points in C such that there is no taxicab segment between r and s in C , but they are connected by some path (drawn by heavy lines in Figure 7.7) that is included in C . Without loss of generality, we can assume that this path is completely outside the rectangular region whose two corners are defined by the points r and s . If we draw horizontal and vertical lines through r and s , they will intersect that path. Let us assume that a horizontal line through s intersects the path at s' and a vertical one through r intersects at r' . Since $s, s' \in A$, the only taxicab line segment between them – that is, the segment ss' – must lie entirely within A . Similarly, $s, s' \in B$ too, so ss' must lie entirely within B . This means that the segment ss' lies entirely within C . According to a similar argument, the segment rr' also lies entirely within C . This means that we can always find at least one taxicab line segment between r and s that lies entirely within C . \square

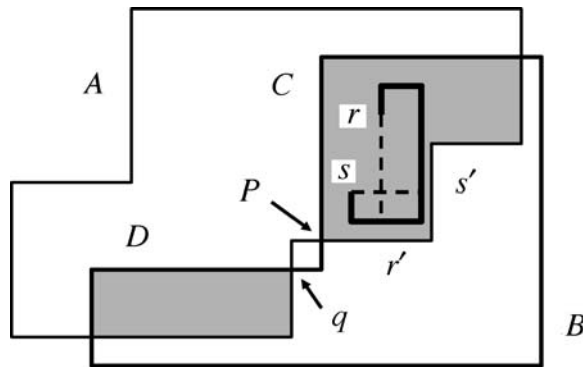


Figure 7.7 The intersection of two taxicab convex polygons

Proposition 7.5. *If A is a WTC polygon, then the Minkowski decomposition $A \ominus B$ is either empty or consists of one or more WTC polygons.*

Proof: The proof follows directly from

$$A \ominus B = \bigcap_{-b \in \vec{B}} A_{-b}$$

and Proposition 7.4. □

Unlike Proposition 7.2, Proposition 7.5 is not exactly the same as the corresponding theorem concerning convex polygons. In the case of conventional convex polygons, Minkowski decomposition $A \ominus B$ is either empty or a single convex polygon – but not more than one polygon.

This difference is reflected in the following result.

Proposition 7.6. *If a WTC polygon S is indecomposable in the WTC polygonal domain, it remains indecomposable even in the connected shape domain. (However, S may be decomposable in the disconnected domain.)*

Proposition 7.6 is analogous to Proposition 7.1 and clearly shows the importance of investigating the WTC domain for indecomposability.

In the proposition, by “connectedness” we informally mean the notion of “consisting of a single piece.” Formally, a set is connected if every pair of elements of the set can be joined by a path that is included in the set. Since all the sets that we consider are implicitly assumed to be closed sets, a “disconnected set” means a set consisting of more than one disjoint pieces.

Proof of Proposition 7.6: Let us assume that there exist two connected sets P and Q – both of which are not WTC polygons – such that $S = P \oplus Q$. The decomposition $S \ominus P$, according to Proposition 7.5, will consist of one or more WTC polygons. But it cannot be empty, since $(S \ominus P) \oplus P$ should be equal to S (see [37]). Let us assume that $S \ominus P = Q_1 \cup Q_2 \cup \dots$, where every Q_i is a WTC polygon. Since $Q \subset S \ominus P$ and Q is a connected set, then Q is necessarily included in one of the components of $S \ominus P$ – say, in Q_1 . Therefore, we shall obtain $P \oplus Q_1 = S$. In a similar way, we can consider the decomposition $S \ominus Q_1$ and argue that we shall obtain a WTC polygon P_1 such that $P_1 \oplus Q_1 = S$. But this contradicts the given condition that S is indecomposable in the WTC domain. This means that our initial assumption is wrong. □

In Figure 7.8(a), we show an example of a WTC polygon S which, since it is decomposable in the connected domain, is also decomposable in the WTC domain. On the other hand, the WTC polygon S in Figure 7.8(b), though decomposable in the general disconnected domain, can be shown to be indecomposable in the WTC domain.

7.3 Computing Minkowski Operations on WTC Polygons

7.3.1 Representation of WTC Polygons

The boundary addition \uplus , which is the kernel of both Minkowski addition and Minkowski decomposition, is a *localized* operation in terms of the outer normal directions. The edge

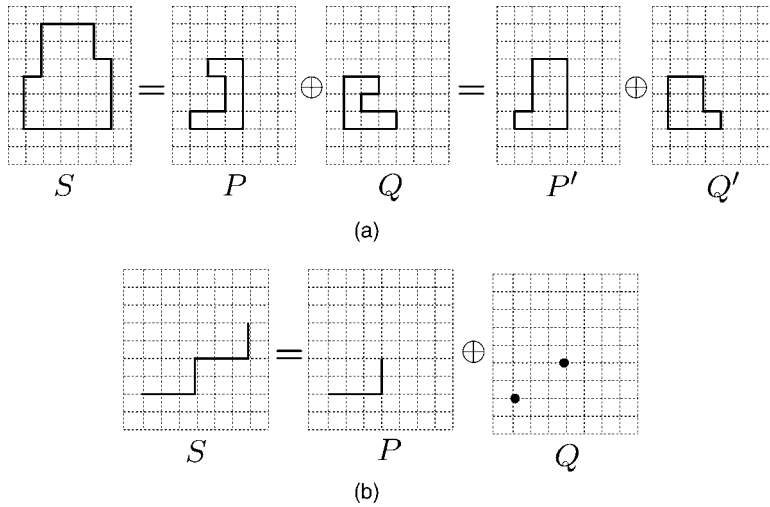


Figure 7.8 Indecomposability in the WTC domain ensures indecomposability in the connected domain, but not in the general disconnected domain: (a) since S is decomposable in the connected domain, it is decomposable in the WTC domain too; S is indecomposable in the WTC domain, but decomposable in the general disconnected domain

points of a polygon subdivide the unit circle of the slope diagram into a number of angular regions. The boundary addition \uplus of the edges of two summand polygons within each angular region always remains confined within that angular region – it does not affect the addition in any other portion. For example, if it becomes possible to partition the slope diagrams of both the summands A and B into the same angular regions as follows,

$$\begin{aligned}\partial A &= (\partial A_{\theta_1-\theta_2}, \partial A_{\theta_2-\theta_3}, \dots, \partial A_{\theta_n-\theta_1}), \\ \partial B &= (\partial B_{\theta_1-\theta_2}, \partial B_{\theta_2-\theta_3}, \dots, \partial B_{\theta_n-\theta_1}),\end{aligned}$$

where $\partial A_{\theta_i-\theta_j}$ denotes the portion of the boundary of A within the angular region between θ_i and θ_j , then according to the slope diagram theory, the boundary addition will turn out to be

$$\begin{aligned}\partial A \uplus \partial B &= (\partial A_{\theta_1-\theta_2} \uplus \partial B_{\theta_1-\theta_2}, \\ &\quad \partial A_{\theta_2-\theta_3} \uplus \partial B_{\theta_2-\theta_3}, \dots, \partial A_{\theta_n-\theta_1} \uplus \partial B_{\theta_n-\theta_1}).\end{aligned}\quad (7.3)$$

Now, for the WTC polygons the first advantage is as follows: *the boundary of every WTC polygon can be partitioned into eight angular regions*; namely, 0° , between 0° and 90° , 90° , between 90° and 180° , 180° , between 180° and 270° , 270° , and between 270° and 360° . For ease of future reference, we denote the four single-direction angular regions by the symbols i_1, i_2, i_3, i_4 , and the four 90° angular regions by the symbols r_1, r_2, r_3, r_4 , respectively (Figure 7.9(a)).

The second advantage is as follows: *in any of these eight angular regions, the boundary portion of the polygon is a single taxicab line segment*. In any i_j -region, the taxicab edge reduces to a vertical or a horizontal line segment. Such an edge can be referred to as an

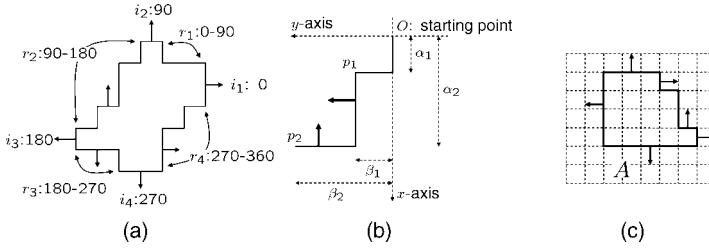


Figure 7.9 The slope diagram representation of a WTC polygon as a hypercomplex number: (a) the partition of a WTC polygon into eight angular regions; (b) representation of an r_j -edge; (c) the slope diagram representation of A as a hypercomplex number $(1, [(1,2), (2,3)], 3, 0, 4, 0, 5, 0)$

i_j -edge. An i_j -edge can be completely specified by its length – say η_j , where η_j denotes a positive integer number. In the case of any r_j -region, the taxicab edge of the polygon will consist of a monotonic chain of “steps,” each step consisting of one vertical and one horizontal line segment. To maintain conformity with the previous notation, such a taxicab edge will be called an r_j -edge and its length will be denoted by the symbol x_j .

Obviously, the specification of x_j , unlike an i_j -edge, needs a sequence of integer numbers rather than a single number. Each step in the r_j -edge can be specified by a pair of integer numbers denoting the lengths of the vertical and the horizontal segments of the step, and x_j will consist of a sequence of such pairs corresponding to the steps in the r_j -edge. We will show that it is more advantageous if the lengths of the vertical and the horizontal segments are all measured from the starting point of the r_j -edge, as shown in Figure 7.9(b). Thus the length x_j can be expressed as

$$x_j = [(\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_n, \beta_n)], \quad (7.4)$$

where α_k and β_k are the lengths of the vertical and the horizontal segments of the k th step of the r_j -edge. All the α 's and β 's are positive integer numbers. Moreover, because of the monotonicity of a taxicab edge, $0 < \alpha_1 < \alpha_2 < \dots < \alpha_n$ and also $0 < \beta_1 < \beta_2 < \dots < \beta_n$. The taxicab length of the r_j -edge is $\alpha_n + \beta_n$. The number of steps n in the r_j -edge is called the *multiplicity* of the edge.

Equation (7.4) indicates that an r_j -edge may be viewed slightly differently. At every r_j -edge, we can build a two-dimensional *local coordinate system* by taking the starting point of the r_j -edge as its origin o , and the vertical and horizontal lines as the coordinate axes. Then any pair (α_k, β_k) in (7.4) will represent the coordinate of the *end-point* p_k of the k th step of the r_j -edge (Figure 7.9(b)). In other words, the r_j -edge can be viewed as a set of points $\{o, p_1, p_2, \dots, p_n\}$, and we can say that

$$x_j \equiv \{o, p_1, p_2, \dots, p_n\}. \quad (7.5)$$

Obviously, since the r_j -edge is a taxicab edge, the points in (7.4) are sorted in both the horizontal and the vertical direction.

(Let us point out here that even an i_j -edge can be viewed in the same way as a set of two points $\{o, v\}$, where v denotes the end-point of the i_j -edge. Clearly, v is a one-dimensional vector, which can be represented by a single number (η_j) , the length of the i_j -edge.)

Once all the η_j 's and x_j 's are specified, the slope diagram of a WTC polygon A can be completely expressed in a compact form, as

$$\partial A \equiv (\eta_1 i_1, x_1 r_1, \eta_2 i_2, x_2 r_2, \dots, \eta_4 i_4, x_4 r_4),$$

which resembles a hypercomplex algebraic number. For further simplification, if we assume that the angular regions will always be considered as a fixed ordered set $(i_1, r_1, i_2, r_2, \dots, i_4, r_4)$, then the above equation can be expressed as an ordered 8-tuple:

$$\partial A \equiv (\eta_1, x_1, \eta_2, x_2, \dots, \eta_4, x_4). \quad (7.6)$$

A typical WTC polygon and its representation as an ordered 8-tuple are shown in Figure 7.9(c). Note that some of the η_j 's or x_j 's may be zero.

7.3.2 The Minkowski Addition of Two WTC Polygons

To determine the Minkowski sum $A \oplus B$ of two WTC polygons whose slope diagrams are given by

$$\begin{aligned} \partial A &\equiv (\eta_1, x_1, \eta_2, x_2, \dots, \eta_4, x_4) \\ \partial B &\equiv (\xi_1, y_1, \xi_2, y_2, \dots, \xi_4, y_4), \end{aligned} \quad (7.7)$$

we can write, from our slope diagram method, that

$$\begin{aligned} \partial A \uplus \partial B &\equiv (\eta_1 \uplus \xi_1, x_1 \uplus y_1, \eta_2 \uplus \xi_2, \\ &\quad x_2 \uplus y_2, \dots, \eta_4 \uplus \xi_4, x_4 \uplus y_4). \end{aligned} \quad (7.8)$$

Note that the boundary addition of two WTC polygons resembles the *component-wise* addition of two hypercomplex numbers representing the summand polygons. It can also be viewed as the component-wise addition of two eight-dimensional vectors or two ordered 8-tuples. But considering it as the addition of two hypercomplex numbers, we intend to indicate that not all of the components are of the same type (the η_j, ξ_j 's are different from the x_j, y_j 's), and therefore we require different addition rules for different types of component.

Since η_j and ξ_j are two positive numbers, $\eta_j \uplus \xi_j$ becomes simply the arithmetic addition of these two numbers; that is,

$$\eta_j \uplus \xi_j = \eta_j + \xi_j. \quad (7.9)$$

The computation of $x_j \uplus y_j$ is a little more involved.

From the viewpoint of slope diagram theory, the simplest x_j, y_j are the r_j -edges whose multiplicity is 1; that is, $x_j = [(\alpha_1, \beta_1)]$ and $y_j = [(\lambda_1, \mu_1)]$ (Figures 7.10(a) and (b)).

From the merged slope diagram, we obtain that $x_j \uplus y_j$ in this case will be the union of two sets of edges; namely, $\{x_j, \lambda_1, -x_j, \mu_1, x_j\}$ and $\{y_j, \alpha_1, -y_j, \beta_1, y_j\}$. It is easy to show that both these sets are, in fact, equivalent and finally yield the same result. (This will also be apparent from the following expansion.) Therefore, we may proceed with only one of these two sets – say, the first one (Figures 7.10(c) and (d)) – and expand in the following way:

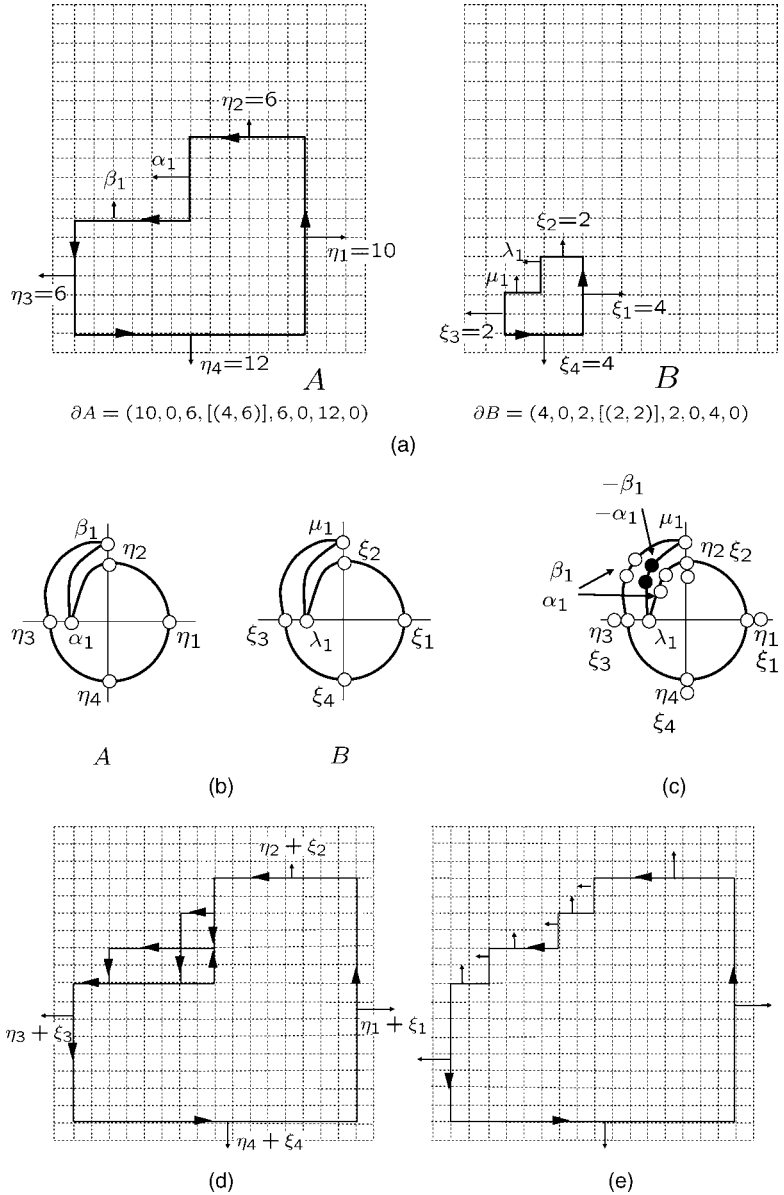


Figure 7.10 The Minkowski addition of two simple WTC polygons: (a) the operand WTC polygons A and B; (b) the slope diagrams of A and B; (c) the merged slope diagram; (d) realization of the merged slope diagram $\partial A \uplus \partial B$; (e) Minkowski addition of A and B, $\text{Pos}(\partial A \uplus \partial B)$

$$\begin{aligned}
x_j \uplus y_j &= \{x_j, \lambda_1, -x_j, \mu_1, x_j\} \\
&= [(\alpha_1, \beta_1), (\alpha_1 + \lambda_1, \beta_1), (\alpha_1 + \lambda_1, 0), (\lambda_1, 0), \\
&\quad (\lambda_1, \mu_1), (\alpha_1 + \lambda_1, \mu_1), (\alpha_1 + \lambda_1, \beta_1 + \mu_1)] \\
&= [(\alpha_1, \beta_1), (\lambda_1, \mu_1), (\alpha_1 + \lambda_1, \beta_1 + \mu_1)].
\end{aligned}$$

Now, if we express x_j and y_j in terms of their end-point sets – that is, as $x_j = \{o, p_1\}$ and $y_j = \{o, q_1\}$ – then the above expression can be written as

$$\begin{aligned}
x_j \uplus y_j &= \{o, p_1, q_1, p_1 + q_1\} \\
&= \{o, p_1\} \oplus \{o, q_1\}.
\end{aligned}$$

The above equation implies that the boundary addition \uplus of two r_j -edges finally reduces to the Minkowski addition \oplus of the corresponding end-point sets. The generalization of the boundary addition $x_j \uplus y_j$ can now be obtained directly. If we consider two general r_j -edges, as

$$x_j = \{o, p_1, p_2, \dots, p_n\}, \quad y_j = \{o, q_1, q_2, \dots, q_m\}, \quad (7.10)$$

then we have

$$\begin{aligned}
x_j \uplus y_j &= \{o, p_1, p_2, \dots, p_n\} \oplus \{o, q_1, q_2, \dots, q_m\} \\
&= \{a + b \mid a \in \{o, p_1, p_2, \dots, p_n\}, \\
&\quad b \in \{o, q_1, q_2, \dots, q_m\}\}.
\end{aligned} \quad (7.11)$$

If (7.11) is expressed in terms of the lengths of the steps, then for

$$\begin{aligned}
x_j &= [(\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_n, \beta_n)], \\
y_j &= [(\lambda_1, \mu_1), (\lambda_2, \mu_2), \dots, (\lambda_m, \mu_m)],
\end{aligned}$$

we can write

$$\begin{aligned}
x_j \uplus y_j &= [(\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_n, \beta_n), \\
&\quad (\lambda_1, \mu_1), (\lambda_2, \mu_2), \dots, (\lambda_m, \mu_m), \dots \\
&\quad (\alpha_1 + \lambda_1, \beta_1 + \mu_1), (\alpha_2 + \lambda_1, \beta_2 + \mu_1), \dots].
\end{aligned} \quad (7.12)$$

Equations (7.8)–(7.12) express the boundary addition $\partial A \uplus \partial B$ of two WTC polygons in simple terms. The next step is to compute $\partial(A \oplus B)$, which is equal to $\text{Pos}(\partial A \uplus \partial B)$, as was described in the previous two chapters and shown in Figure 7.10(e). We shall see that this computation turns out to be straightforward for the following facts:

- The self-crossings in $\partial A \uplus \partial B$ are due to the r_j -edges in the summands, but not due to the i_j -edges.
- The self-crossings are completely *localized* in the following sense. An $(x_j \uplus y_j)$ does not intersect any other edge of the boundary sum $\partial A \uplus \partial B$.
- $\text{Pos}(\partial A \uplus \partial B) = \partial(A \oplus B)$ is a WTC polygon (see Proposition 7.2).

As a result, every i_j -edge of $\partial(A \oplus B)$ is the same as the corresponding i_j -edge of $\partial A \uplus \partial B$, and every r_j -edge of $\partial(A \oplus B)$ can be obtained solely from the corresponding $(x_j \uplus y_j)$ in the following way:

1. Sort the pairs $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_n, \beta_n), (\lambda_1, \mu_1), (\lambda_2, \mu_2), \dots, (\lambda_m, \mu_m), \dots, (\alpha_1 + \lambda_1, \beta_1 + \mu_1), (\alpha_2 + \lambda_1, \beta_2 + \mu_1), \dots$ obtained from (7.12) in ascending order according to the first elements of the pairs.
2. Sort the resulting ordered pairs, in ascending order, according to the second elements of the pairs. (To go to this step, you require both “merging” and “sorting.”)

The merging and sorting, for two pairs $(\lambda_1, \mu_1), (\lambda_2, \mu_2)$ can be done according to the following rule:

- (a) if $\gamma_1 < \gamma_2$ and $\delta_1 < \delta_2$, then $(\gamma_1, \delta_1), (\gamma_2, \delta_2) = (\gamma_1, \delta_1), (\gamma_2, \delta_2)$;
- (b) if $\gamma_1 \leq \gamma_2$ and $\delta_1 \geq \delta_2$, then $(\gamma_1, \delta_1), (\gamma_2, \delta_2) = (\gamma_1, \delta_1)$;
- (c) if $\gamma_1 \geq \gamma_2$ and $\delta_1 \leq \delta_2$, then $(\gamma_1, \delta_1), (\gamma_2, \delta_2) = (\gamma_2, \delta_2)$;
- (d) if $\gamma_1 > \gamma_2$ and $\delta_1 > \delta_2$, then $(\gamma_1, \delta_1), (\gamma_2, \delta_2) = (\gamma_2, \delta_2), (\gamma_1, \delta_1)$.

The rationale behind the rule can be clearly understood from Figure 7.11. In fact, the r_j -edge of $\partial(A \oplus B)$ is simply the shortest taxicab edge through the points $\{o, p_1, p_2, \dots, p_n\} \oplus \{o, q_1, q_2, \dots, q_m\}$. In this chapter, however, we shall not go into such details, but instead present an example to demonstrate the entire method that we have described so far.

Example 7.1. (See Figure 7.12.) Let

$$\partial A = (4, 0, 0, [(1, 2), (3, 4)], 1, 0, 4, 0)$$

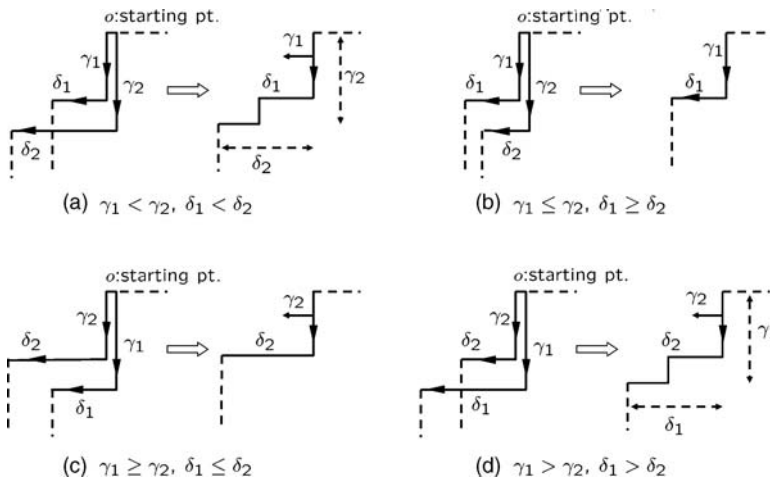


Figure 7.11 Determination of the r_j -edge of the Minkowski sum from the corresponding $x_j \uplus y_j$ – the process needs sorting and merging: (a) $\gamma_1 < \gamma_2, \delta_1 < \delta_2$; (b) $\gamma_1 \leq \gamma_2, \delta_1 \geq \delta_2$; (c) $\gamma_1 \geq \gamma_2, \delta_1 \leq \delta_2$; (d) $\gamma_1 > \gamma_2, \delta_1 > \delta_2$

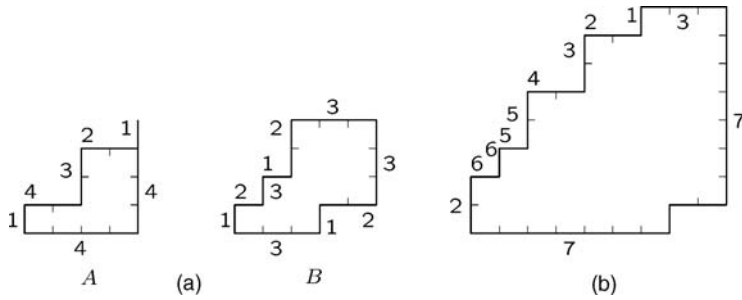


Figure 7.12 Determination of the Minkowski sum of two WTC polygons by means of addition, sorting, and merging of integers: the summand WTC polygons A and B ; (b) the Minkowski sum of A and B

and

$$\partial B = (3, 0, 3, [(2, 1), (3, 2)], 1, 0, 3, [(1, 2)]).$$

By component-wise addition, we obtain

$$\partial A \uplus \partial B = (7, 0, 3, x_2 \uplus y_2, 2, 0, 7, [(1, 2)]),$$

where (by using (7.12))

$$\begin{aligned} x_2 \uplus y_2 &= [(1, 2), (3, 4)] \uplus [(2, 1), (3, 2)] \\ &= [(1, 2), (3, 4), (2, 1), (3, 2), (3, 3), (4, 4), (5, 5), (6, 6)]. \end{aligned}$$

To obtain $\text{Pos}(\partial A \uplus \partial B)$, we have to simplify only the $x_2 \uplus y_2$ -edge of the above expression:

$$\begin{aligned} &(1, 2), (3, 4), (2, 1), (3, 2), (3, 3), (4, 4), (5, 5), (6, 6) \\ &= \underbrace{(1, 2), (2, 1), (3, 4), (3, 2)}_{\text{(sorted by first elements)}}, (3, 3), (4, 4), (5, 5), (6, 6) \\ &= (1, 2), \underbrace{(3, 4), (3, 3), (4, 4)}_{\text{(sorted and merged)}}, (5, 5), (6, 6) \\ &= (1, 2), \underbrace{(3, 4), (4, 4)}_{\text{(sorted and merged)}}, (5, 5), (6, 6) \\ &= (1, 2), (3, 4), (5, 5), (6, 6) \\ &\quad \text{(already sorted by second elements).} \end{aligned}$$

□

7.3.3 The Minkowski Decomposition of Two WTC Polygons

Minkowski decomposition $A \ominus B$ is considered briefly in what follows. If the slope diagram of the operands are given by

$$\partial A = (\eta_1, x_1, \eta_2, x_2, \dots, \eta_4, x_4)$$

$$\partial B = (\xi_1, y_1, \xi_2, y_2, \dots, \xi_4, y_4),$$

then

$$\partial B^{-1} = (-\xi_1, -y_1, -\xi_2, -y_2, \dots, -\xi_4, -y_4).$$

Therefore,

$$\begin{aligned} \partial A \uplus \partial B^{-1} = & (\eta_1 \uplus (-\xi_1), x_1 \uplus (-y_1), \eta_2 \uplus (-\xi_2), \\ & x_2 \uplus (-y_2), \dots, \eta_4 \uplus (-\xi_4), x_4 \uplus (-y_4)), \end{aligned} \quad (7.13)$$

which is simply the *component-wise subtraction* of two hypercomplex numbers, ∂A and ∂B .

Just like the Minkowski addition case, we can show that $\eta_j \uplus (-\xi_j)$ boils down to the arithmetic subtraction of two integer numbers; that is, $\eta_j \uplus (-\xi_j) = \eta_j - \xi_j$. The computation of $x_j \uplus (-y_j)$ can also be carried out in a similar way. For example, if $x_j = [(\alpha_1, \beta_1)]$ and $y_j = [(\lambda_1, \mu_1)]$, then

$$\begin{aligned} x_j \uplus (-y_j) &= \{-y_j, \alpha_1, -(-y_j), \beta_1, -y_j\} \\ &= [(-\lambda_1, -\mu_1), (\alpha_1 - \lambda_1, -\mu_1), (\alpha_1, 0), \\ &\quad (\alpha_1, \beta_1), (\alpha_1 - \lambda_1, \beta_1 - \mu_1)]. \end{aligned}$$

Even if the multiplicities of the r_j -edges of the summands are greater than one, we may proceed in a similar way. An example of Minkowski decomposition is shown in Figure 7.13.

However, the next step – that is, to compute $\text{Pos}(\partial A \uplus \partial B^{-1})$ – is not as straightforward as the addition case for the following reasons:

- The self-crossings due to $x_j \uplus (-y_j)$ are not localized. This means that an $(x_j \uplus (-y_j))$ may intersect other edges of the boundary sum $\partial A \uplus \partial B^{-1}$ (see Figure 7.13(b)).
- $\eta_j - \xi_j$ may become negative, and thereby give rise to self-crossing edges.

Although it is possible to devise some ingenious methods to compute $\text{Pos}(\partial A \uplus \partial B^{-1})$, in this chapter we shall not concern ourselves with such a task.

7.4 A Few Results on Indecomposability in the WTC Domain

7.4.1 The Number of Indecomposable Shapes

How many indecomposable shapes are there? We will show that there are infinitely many. To emphasize the resemblance between indecomposable shapes and prime numbers, we provide *a proof that is analogous to Euclid's proof* that there are infinitely many primes. For our final proof, we need a couple of definitions and the following lemma.

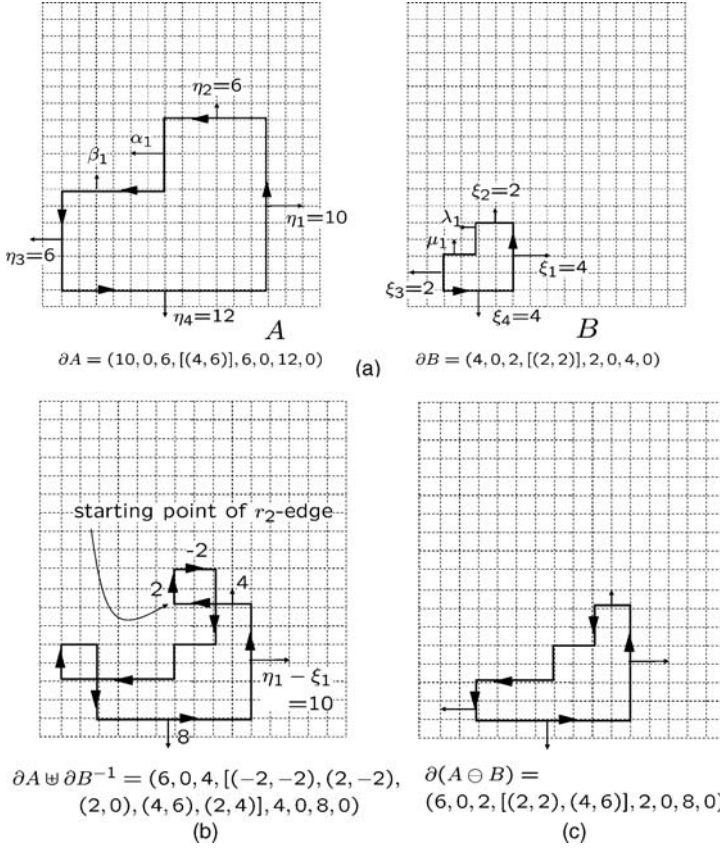


Figure 7.13 The Minkowski decomposition of two WTC polygons: (a) the operand WTC polygons A and B ; (b) the boundary sum of A and B^{-1} ; (c) the Minkowski decomposition of A and B

Definition 7.4 (gap): The gap between two point sets, A and B , is defined as

$$\text{gap}(A, B) = \inf_{a \in A, b \in B} (\rho_E(a, b)),$$

where $\rho_E(a, b)$ denotes the Euclidean distance between the points a and b . □

Definition 7.5 (diameter): The diameter of a point set A is defined as

$$\text{diam}(A) = \sup_{a_1, a_2 \in A} (\rho_E(a_1, a_2)).$$

(In an informal sense, the symbols \inf and \sup may be read as “min” and “max,” respectively.) □

Lemma 7.7. *Let the gap(A, B) between two point sets A and B be larger than the diam(C) of another point set C . Then*

$$(A \cup B) \ominus C = (A \ominus C) \cup (B \ominus C).$$

Proof: If $x \in (A \cup B) \ominus C$, then $C_x \subset (A \cup B)$. Now, the $\text{diam}(C_x) < \text{gap}(A, B)$, and so is necessarily included either in A or in B . Hence $(A \cup B) \ominus C = (A \ominus C) \cup (B \ominus C)$. \square

Proposition 7.8. *There are infinitely many indecomposable shapes.*

Proof: Suppose that there are only finitely many indecomposable shapes I_1, I_2, \dots, I_k . None of these I_j 's are assumed to be a singleton point, which is like a unit element. Let S denote their Minkowski sum; that is,

$$S = I_1 \oplus I_2 \oplus \dots \oplus I_k.$$

Now add (in the sense of set union) a singleton point set $\{p\}$ to S in such a way that $\text{gap}(S, \{p\})$ is larger than $\text{diam}(I_j)$, $j = 1, \dots, k$. Let this set be denoted by S_1 ; that is,

$$S_1 = S \cup \{p\}.$$

According to Lemma 7.7, we can write

$$S_1 \ominus I_j = (S \ominus I_j) \cup (\{p\} \ominus I_j).$$

Since any I_j contains more than one point, $\{p\} \ominus I_j = \emptyset$. This means that

$$S_1 \ominus I_j = S \ominus I_j,$$

and, therefore,

$$(S_1 \ominus I_j) \oplus I_j = (S \ominus I_j) \oplus I_j.$$

Since I_j is a summand of S , therefore $(S \ominus I_j) \oplus I_j = S$ (see [37]). Since $S_1 \neq S$, we obtain, $(S_1 \ominus I_j) \oplus I_j \neq S_1$. This implies that I_j is not a summand of S_1 . In other words, none of I_1, I_2, \dots, I_k can be a summand of S_1 . This implies that there must be some other indecomposable shape that can be a summand of S_1 , or that S_1 itself may be an indecomposable shape. \square

7.4.2 Identifying Indecomposable Polygons within the WTC Domain

We shall first show that if it becomes possible to identify one indecomposable shape, then a class of indecomposable shapes can be immediately identified by a set of geometric transformations.

Proposition 7.9. *The indecomposability or decomposability of a shape is invariant under every affine transformation.*

The proposition essentially states that if some point set S is indecomposable, then its transformed image $\mathcal{A}(S)$ under an affine transformation \mathcal{A} will also be indecomposable. Similarly, if S is decomposable, $\mathcal{A}(S)$ will be decomposable too.

Proof of Proposition 7.9: In two dimensions, an affine transformation $\mathcal{A}(p = (x, y))$ is defined as

$$\begin{aligned}x' &= \chi_1 x + \phi_1 y + \kappa_1, \\y' &= \chi_2 x + \phi_2 y + \kappa_2,\end{aligned}$$

where χ_i and so on are real numbers such that $\chi_1 \phi_2 - \chi_2 \phi_1 \neq 0$. If the translation (κ_1, κ_2) is not considered, then an affine transformation is called a *linear transformation* \mathcal{L} . It immediately follows from the definition of Minkowski addition that, $\mathcal{L}(A \oplus B) = \mathcal{L}(A) \oplus \mathcal{L}(B)$. If (κ_1, κ_2) is considered, we do not get such a strict equality. But with the assumption that all the translates of a point set are equivalent, we can write $\mathcal{A}(A \oplus B) = \mathcal{A}(A) \oplus \mathcal{A}(B)$.

Let us now assume that a set S is indecomposable, but one of its affine transformed images – say, $\mathcal{A}(S)$ – is decomposable; that is, $\mathcal{A}(S) = P \oplus Q$. The inverse \mathcal{A}^{-1} of \mathcal{A} is also an affine transformation. Therefore, $\mathcal{A}^{-1}(\mathcal{A}(S)) = \mathcal{A}^{-1}(P) \oplus \mathcal{A}^{-1}(Q)$, or, $S = \mathcal{A}^{-1}(P) \oplus \mathcal{A}^{-1}(Q)$. This contradicts the assumption that S is indecomposable. In an exactly similar way, we can argue for a decomposable S . \square

Most of the familiar geometric transformations – namely, translation, rotation, reflection, and similarity transformation – are affine transformations. Thus the proposition implies that every rotated, reflected, or scaled image of an indecomposable shape is also indecomposable. For example, in the indecomposability question, a WTC polygon $(\eta_1, x_1, \eta_2, x_2, \dots, \eta_4, x_4)$ is equivalent to $(\eta_4, x_4, \eta_1, x_1, \dots, \eta_3, x_3)$, which is obtained by circularly shifting the elements by two places. The shifting specifies a 90° rotation.

In the next few propositions, we identify some of the WTC polygons that are indecomposable in the WTC domain. We take the following approach. Assuming that a WTC polygon is given in the form of a hypercomplex number

$$\partial S = (\sigma_1, z_1, \sigma_2, z_2, \dots, \sigma_4, z_4),$$

we try to find two numbers

$$\begin{aligned}\partial A &= (\eta_1, x_1, \eta_2, x_2, \dots, \eta_4, x_4), \\ \partial B &= (\xi_1, y_1, \xi_2, y_2, \dots, \xi_4, y_4),\end{aligned}$$

such that (a) $\partial S = \text{Pos}(\partial A \uplus \partial B)$ and (b) the numbers ∂A and ∂B both represent physically realizable WTC polygons. If we can show that no such $\partial A, \partial B$ exist, then the given polygon S is reported as indecomposable in the WTC domain.

The first in this series of propositions is obvious.

Proposition 7.10. *Any WTC polygon resulting from a 2-point binary image is indecomposable.*

The WTC polygons resulting from 2-point binary images are simply a horizontal or a vertical line segment of unit length (Figure 7.14(a)). We call them *L-polygons*. (We find that *L-polygons* behave very much like the number 2.)

The class of WTC polygons that we consider next are like simple triangles in the WTC domain. A few such polygons are shown in Figure 7.14(b). The following definition can be used to denote this class of polygons.

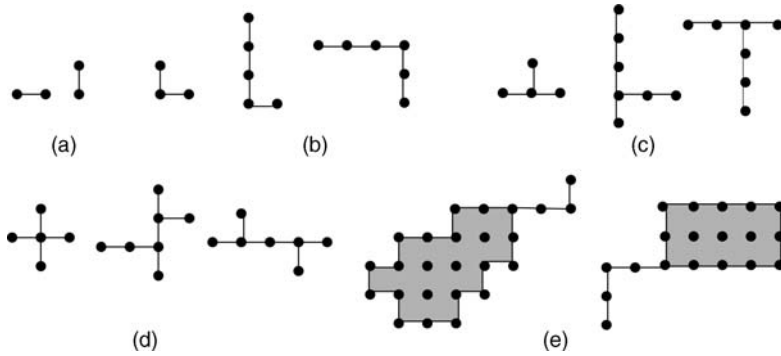


Figure 7.14 A few sets of indecomposable WTC polygons: (a) L -polygons; (b) a few examples of R_1 -polygons; (c) a few R_2 -polygons; (d) a few R_4 -polygons; (e) a few GR_1 -polygons

Definition 7.6 (R_1 -polygon): A WTC polygon is called an R_1 -polygon if it has the following characteristics: (i) it has only one r -edge, (ii) the multiplicity of that r -edge is 1, and (iii) both of the i -edges adjacent to that nonzero r -edge have length 0. \square

You may have noted that a R_1 -polygon can have only three taxicab edges – one r -edge and two i -edges.

Our proposition concerning R_1 -polygons requires the following lemma.

Lemma 7.11. *If a WTC polygon A has an r_j -edge of multiplicity 1 and the multiplicity of the same r_j -edge of another WTC polygon B is also 1, then the corresponding r_j -edge of the Minkowski sum $A \oplus B$ must have multiplicity more than 1.*

Proof: It immediately follows from the Minkowski addition procedure (see Section 7.3.2) that the multiplicity of the sum will be either 2 or 3. \square

Proposition 7.12. *Every R_1 -polygon is indecomposable.*

Proof: Any R_1 -polygon consists of an r_j -edge of multiplicity 1, and the i_{j+2} -edge and the i_{j+3} -edge. Let us consider a typical R_1 -polygon S that is expressed as $\partial S = (0, z_1, 0, 0, \sigma_3, 0, \sigma_4, 0)$, where $z_1 = [(\sigma_4, \sigma_3)]$ (see the first or the second polygon in Figure 7.14(b)). Assume that S can be expressed as the Minkowski sum of two polygons $\partial A = (0, x_1, 0, 0, \eta_3, 0, \eta_4, 0)$, and $\partial B = (0, y_1, 0, 0, \xi_3, 0, \xi_4, 0)$. Since z_1 has multiplicity 1, then according to Lemma 7.11 we cannot obtain z_1 from $x_1 \uplus y_1$ unless one of them, say y_1 is equal to zero. Therefore, x_1 becomes equal to $z_1 = [(\sigma_4, \sigma_3)]$, and the lengths η_3 and η_4 of the nonzero i -edges of ∂A become equal to σ_3 and σ_4 , respectively. This means that ∂A becomes exactly equal to ∂S , and that ∂B becomes a singleton point. So S is indecomposable.

We can now consider every affine variation of S , and using Proposition 7.9 we can argue that every R_1 -polygon is indecomposable. \square

The next result is essentially a generalization of Proposition 7.12. We consider a class of WTC polygons having the following characteristics: (i) the multiplicity of every r -edge in a

polygon is 1, and (ii) both of the i -edges adjacent to a nonzero r -edge have length 0. It can immediately be seen that this class of polygons can be subdivided into three subclasses of WTC polygons:

1. R_1 -polygons (one r -edge). $(0, z_1, 0, 0, \sigma_3, 0, \sigma_4, 0)$, and affine versions of it; see Figure 7.14(b).
2. R_2 -polygons (two r -edges). $(0, z_1, 0, z_2, 0, 0, \sigma_4, 0)$, and affine versions of it; see Figure 7.14(c).
3. R_4 -polygons (four r -edges). $(0, z_1, 0, z_2, 0, z_3, 0, z_4)$, and affine versions of it; see Figure 7.14(d).

Note that no R_3 -polygon – that is, WTC polygon having three such r -edges – can exist physically.

Proposition 7.13. *If every r -edge of a WTC polygon has multiplicity 1 and both the i -edges adjacent to every nonzero r -edge have length 0, then the polygon is indecomposable.*

Proof: A polygon having the above characteristics can be an R_1 -, R_2 -, or R_4 -polygon. According to Proposition 7.12, every R_1 -polygon is indecomposable.

Let us assume that the given polygon S is an R_4 -polygon. If S is equal to $A \oplus B$, then according to Lemma 7.11 the summands ∂A and ∂B cannot have the same r_j -edges; that is, if ∂A has an r_j -edge and ∂B has an r_k -edge, then $j \neq k$. This means that one of the summands has either one r -edge or two r -edges, and no i -edge at all. Obviously, such a WTC polygon cannot be physically realized. Therefore, every R_4 -polygon is indecomposable.

Arguing in a similar way, we can show that every R_2 -polygon is also indecomposable. \square

The R_1 -, R_2 -, and R_4 -classes of polygons can be considered to be the basic indecomposable classes in the WTC domain. We shall now show that, by means of these basic classes, we can define other classes of indecomposable polygons. Consider the following definition.

Definition 7.7 (GR_1 -polygon): A WTC polygon formed by gluing an R_1 -polygon to another polygon is called a GR_1 -polygon (as shown in Figure 7.14(e)). \square

Here, “gluing” means a restricted form of set union where one end-point of the R_1 -polygon intersects the other polygon at a single point.

Proposition 7.14. *Every GR_1 -polygon is indecomposable.*

Proof: A typical GR_1 -polygon S can be expressed as

$$\partial S = (\sigma_1, 0, 0, [(\sigma_1, \delta_1), \dots], \sigma_3, z_3, \sigma_4, z_4).$$

If S is assumed to be the sum of two WTC polygons A and B , then the general expressions of the summands will be as follows:

$$\partial A = (\eta_1, 0, 0, [(\alpha_1, \beta_1), \dots], \eta_3, x_3, \eta_4, x_4),$$

$$\partial B = (\xi_1, 0, 0, [(\lambda_1, \mu_1), \dots], \xi_3, y_3, \xi_4, y_4).$$

Obviously, for the physical realization of A and B , it is required that $\alpha_1 \leq \eta_1$ and $\lambda_1 \leq \xi_1$.

Since $S = A \oplus B$, $\sigma_1 = \eta_1 + \xi_1$. Furthermore, the r_2 -edge of ∂S – that is, $[(\sigma_1, \delta_1), \dots]$ – has to be obtained by adding the r_2 -edges of the summands. This demands that either α_1 or λ_1 should be equal to σ_1 . Let us assume that $\alpha_1 = \sigma_1$. This means that $\sigma_1 \leq \lambda_1$. But σ_1 is the sum of η_1 and ξ_1 , both of which are positive numbers. Therefore, $\sigma_1 = \eta_1$, and $\xi_1 = 0$. Therefore, $\lambda_1 \leq 0$, which means that $\lambda_1 = 0$. Therefore, ∂B now becomes equal to $(0, 0, 0, 0, \xi_3, y_3, \xi_4, y_4)$. Obviously, such a WTC polygon cannot be realized in practice. So the given S is indecomposable. \square

In a similar way, we can consider various other classes of WTC polygons and examine their indecomposability properties. In this exposition, we have decided not to take this investigation any further. Our basic intention in stating this set of propositions (Propositions 7.10–7.14) is to demonstrate how we can determine whether or not a given class of WTC polygons is indecomposable in a purely *number-theoretic* way.

Indeed, the same number-theoretic technique can be employed if, instead of a class of WTC polygons, we are given a single WTC polygon. In addition to determining the indecomposability characteristic of the given polygon, we can also find a set of summands of the polygon if the polygon turns out to be a decomposable one. To elucidate the point, we will present here one illustrative example.

Example 7.2. (See Figure 7.15.) Let the following WTC polygon be given:

$$\partial S = (4, [(4, 2)], 0, [(3, 3)], 3, [(3, 1)], 0, [(1, 4)]) .$$

It is evident that the polygon S is a decomposable one, since both its i_1 -edge and its i_3 -edge are nonzero. A physically realizable WTC polygon can be formed immediately by means of these two edges, and we can decompose S as follows:

$$\begin{aligned} \partial S &= (3, 0, 0, 0, 3, 0, 0, 0) \\ &\quad \uplus (1, [(4, 2)], 0, [(3, 3)], 0, [(3, 1)], 0, [(1, 4)]) . \end{aligned}$$

This decomposition is shown in Figure 7.15(a), where the two summands are referred to as A and B , respectively.

The first summand A can be further decomposed into three vertical line segments of unit length (which we did not show in the figure). However, we cannot immediately say whether or not the other summand B can be decomposed further. Let us start with one of its r -edges – say, with the r_1 -edge of B . The physically realizable WTC polygon B' that has only one r -edge equal to that r_1 -edge will take the following form: $\partial B' = (0, [(4, 2)], 0, 0, 2, 0, 4, 0)$. Obviously, B' cannot be a summand of B , since its i_3 -edge and its i_4 -edge are nonzero (Figure 7.15(b)). To force the i_3 -edge to be equal to zero, we have to introduce an r_2 -edge which, in this case, must be equal to the r_2 -edge of B . The resulting WTC polygon then becomes: $\partial B'' = (0, [(4, 2)], 0, [(3, 3)], 0, 0, 3, [(1, 4)])$. Clearly, B'' also cannot be a summand of B , since its i_4 -edge is nonzero. To make the i_4 -edge to be equal to zero, we therefore introduce the r_3 -edge, which must be equal to the r_3 -edge of B . This results in the following polygon: $\partial B''' = (0, [(4, 2)], 0, [(3, 3)], 0, [(3, 1)], 0, [(2, 4)])$. B''' cannot be a summand of B , since its r_4 -edge is not equal to the r_4 -edge of B . Now, the only way in which we can transform B''' is to make it equal to B . This means that B is indecomposable. The polygons B' , B'' , B''' are all shown in Figure 7.15(b).

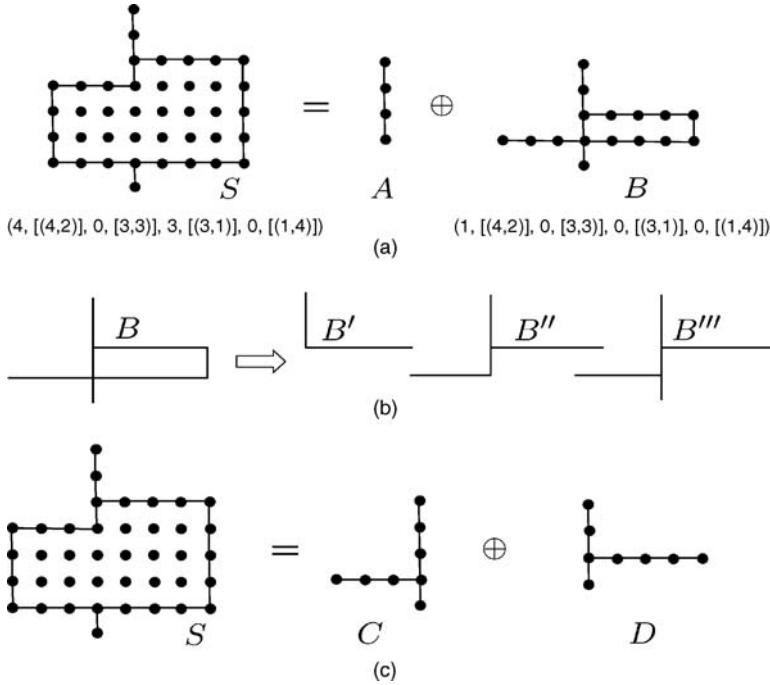


Figure 7.15 A typical WTC polygon S and its decompositions by means of our number-theoretic technique: (a) a typical WTC polygon S and one of its decompositions as a Minkowski sum; (b) in an attempt to decompose B , we are forming components of increasing complexity; (c) another decomposition of S as a Minkowski sum of indecomposable shapes

The given polygon S can be decomposed in other ways too. Approaching the problem in exactly the same way, we obtain a decomposition that is shown in Figure 7.15(c). Note that the summand C and D cannot be decomposed further, since both of them turn out to be R_2 -polygons. In the next subsection, we will briefly discuss other kinds of indecomposability test, which are more computational in nature. \square

7.4.3 Simple Indecomposability Tests

The simplest primality test is the *trial division*: take an odd integer m and check whether or not it divides the given number n . If m divides n , then n is composite; try with all m from 3 to \sqrt{n} .

In a similar spirit, to determine whether or not a given point set S is indecomposable, the simplest *indecomposability test* would be to take a point set B and check whether or not B can be a summand of S . This test can easily be incorporated by using the following result.

Proposition 7.15. *A point set B is a summand of a point set S , iff*

$$(S \ominus B) \oplus B = S.$$

The proof of this proposition can be found in [90] or [37].

(Here, we can point out the striking resemblance between the above geometric proposition and the number-theoretic proposition that states that a positive integer m is a divisor of a positive integer n iff $\lfloor n/m \rfloor \cdot m = n$; the floor function notation $\lfloor x \rfloor$ means the greatest integer less than or equal to x .)

An *algorithm* to test the indecomposability characteristic of a given binary image S follows immediately from Proposition 7.15. Let the size of S be $n_1 \times n_2$. Consider every binary image B whose size is less than $n_1 \times n_2$, and check whether or not $(S \ominus B) \oplus B$ is equal to S . If for any of these B 's the equality holds, then S is not an indecomposable image and that particular B is a summand of S .

For the WTC polygons, the computations of $S \ominus B$ and then $(S \ominus B) \oplus B$ can be carried out by using the algorithms described in Sections 7.3.3 and 7.3.2, respectively.

Interestingly, when both S and B are WTC polygons, testing whether or not B is a summand of S can be accomplished more directly. To express this idea, we need to introduce the following notion.

Definition 7.8 (less than or equal to): Let L_1 and L_2 be two taxicab line segments. We say that L_1 is *less than or equal to* L_2 if there exists a taxicab line segment L_x , such that L_2 can be generated by Minkowski addition of L_1 and L_x . If L_x turns out to be a single point, we say that L_1 is *equal to* L_2 . \square

It follows from the above definition that, if L_1 and L_2 are both vertical line segments or horizontal line segments, then L_1 is less than or equal to L_2 depending on whether the length of L_1 is less than or equal to that of L_2 .

Proposition 7.16. *A WTC polygon B cannot be a summand of a WTC polygon S if every i_j -edge and r_j -edge of B is not less than or equal to the corresponding i_j -edge and r_j -edge of S .*

The proof follows directly from the discussion in Section 7.3.2. Note the resemblance between this proposition and the corresponding proposition regarding convex polygons.

7.5 A Brief Summing Up

Let us briefly summarize the important concepts introduced and developed in this chapter:

1. *The WTC Domain.* The identification of the WTC domain is significant for a number of reasons. First, the behavior of a WTC image in the binary image domain is very similar to that of a convex polygon in the real Euclidean domain; most of the theorems proved for convex polygons can be shown to hold in the WTC domain too. Second, it clearly demonstrates that the notion of convexity is more general than the conventional notion of convexity in real Euclidean space. Here, just by “appropriately” choosing a different distance function, we have been able to identify the WTC domain from a generalized concept of convexity. Independent of its importance in the indecomposability problem, the WTC domain itself demands further attention.
2. *Efficient Algorithms for Minkowski Addition and Decomposition.* As a by-product of our investigation, we obtain very efficient methods of computing Minkowski addition and decomposition of binary images. Any binary image can be thought of as a union of a

- few WTC images (just as a nonconvex object is often treated as a union of convex objects), and thereby their Minkowski operations finally reduce to Minkowski operations on WTC polygons. The latter operations, as we have shown, are essentially simple additions/subtractions of hypercomplex numbers.
3. *The Indecomposability Problem.* The importance of the indecomposability problem in morphology has not, we feel, been fully recognized up to the present day. We have attempted to show that, apart from its practical relevance, the intrinsic mathematical importance of the problem in shape theory can be compared with the primality problem in number theory.

Far from marking the end of an investigation into the indecomposability of binary images, this work signals a host of new perspectives and questions. Let us briefly mention a few such questions, which must be taken into consideration in the near future.

7.5.1 Why Does the Uniqueness of Shape Decomposition Fail?

One of the basic differences that distinguishes the theory of indecomposable shapes from that of prime numbers is the question of *uniqueness* in decomposition. The *fundamental theorem of arithmetic* states that the factoring of any positive integer n into primes is unique apart from the order of the prime factors. But any analogous theorem concerning geometric shapes fails to hold, in general. In the WTC domain, for example, we can show that some polygons can be decomposed in more than one way as Minkowski sums of indecomposable shapes (Figure 7.16).

We must, however, note that we are simply fortunate that in the integer number domain \mathbb{Z} the unique factorization property holds, because many other number systems exist in which factorization is not unique. We present two simple examples: (1) Consider the set E of all positive integers; that is, the set $2, 4, 6, 8, \dots$. E is a multiplicative system, and hence the product of any two elements of E is also a member of E . In this system, the primes are $2, 6, 10, 14, \dots$, whereas $4, 8, 12, \dots$ are the “composite numbers.” Now, the number 60 has two factorings into primes; namely, $60 = 2 \cdot 30 = 6 \cdot 10$. (2) Consider the set Q of numbers $m + n\sqrt{-6}$, where m, n range over all integers. This is also a multiplicative system. In this system, it can be shown that the numbers $2, 5, 2 + \sqrt{-6}$ and $2 - \sqrt{-6}$ are all primes. Now, the number 10 has two factorings: $10 = 2 \cdot 5 = (2 + \sqrt{-6}) \cdot (2 - \sqrt{-6})$.

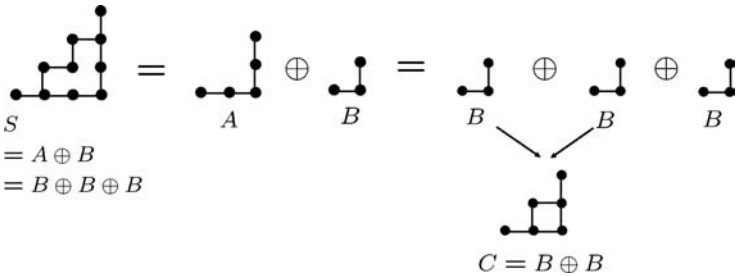


Figure 7.16 A WTC polygon can be decomposed as the sum of indecomposable shapes in more than one way

The point that we want to make by means of these two examples is as follows. If the *decomposition uniqueness* does not hold in some shape domain, by “appropriately” *enlarging* or *restricting* the shape domain, such uniqueness can be achieved. For example, in the *strongly taxicab convex* domain (which consists of horizontal line segments, vertical line segments, squares, and rectangles), it appears that such a uniqueness holds. The question, therefore, is: “What is the necessary condition that the elements of a shape domain must satisfy to achieve decomposition uniqueness within that domain?”

7.5.2 How Many Indecomposable Shapes are There?

We have already seen that there are infinitely many indecomposable shapes. The next question is: “How many indecomposable images are there whose size is less than or equal to $n_1 \times n_2$?”

A somewhat related question is: “Given a binary image of size $n_1 \times n_2$, what are the upper and lower bounds of the number of indecomposable summands needed to express the image?”

7.5.3 How Can We Define New Equivalence Classes of Polygons?

It is known that in the real Euclidean convex domain, if $A \oplus B = C \oplus B$, then $A = C$. But such a theorem does not hold in the WTC domain. We have already shown an example in Figure 7.16 where $A \oplus B = C \oplus B$, but $A \neq C$. From the viewpoint of taxicab edges, however, A and C can be considered as equivalent. Both of the polygons have the same i -edges, whose lengths are equal (two units each), and also the r_2 -edges of both polygons have exactly the same length (four units). Therefore, we can say that $A \equiv C$. A and C are not *congruent*, since the multiplicities of their r_2 -edges are different. The question, therefore, is: “How can we define some new equivalence relation among the WTC polygons so that all the results of the real Euclidean convex domain hold true even in the WTC domain?”

7.5.4 Can We Devise Laws of Exponents, and Eventually Binomial Formulas for Shapes?

The last example (Figure 7.16) is instructive in another interesting way. Keeping in mind the analogy between multiplication and Minkowski addition, let us denote $B \oplus B \oplus \dots \oplus B$ (n terms) by the symbol $B^{\oplus n}$. The example in Figure 7.16 can, therefore, be expressed as $B^{\oplus 3} = A \oplus B$, and we note that $A \equiv B^{\oplus 2}$ since $C = B^{\oplus 2}$. Its resemblance to the number-theoretic equation $b^3 = a \cdot b$ (where a, b are integer or real numbers), and thereby $a = b^2$, will certainly prompt us to investigate this further. Let P and A be two binary images such that

$$P \oplus A^{\oplus m} = A^{\oplus n}, \quad (7.14)$$

where m and n are positive integers. In this case, we can consider P to be equivalent to the image $A^{\oplus(n-m)}$; that is, $P \equiv A^{\oplus(n-m)}$. Geometrically, this equivalence implies that in a “certain sense” the shapes of P and $A^{\oplus(n-m)}$ are the same, although P and $A^{\oplus(n-m)}$ may not be congruent. Let us denote any member of this equivalence class by the symbol $A^{n-m} = A^k$, where $k = n - m$. For a simple binary image A , some of the members of A^2 and A^3 are shown in Figure 7.17. (If m is the smallest positive integer satisfying (7.14), then we may consider m as a measure of the *defect* of an image with respect to the base image A .)

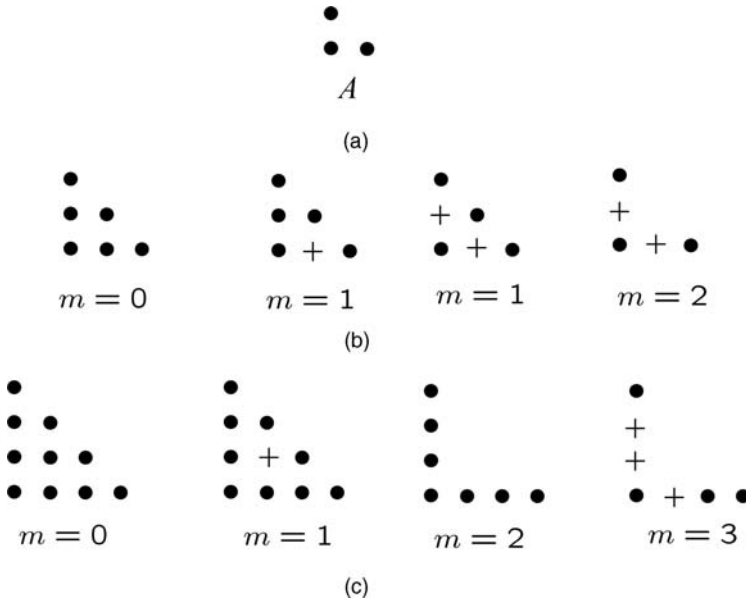


Figure 7.17 A simple WTC polygon A and a few members of the A^2 and A^3 classes: (a) a typical binary image A ; (b) some of the members of A^2 with different defects m ; (c) some of the members of A^3 with various defects

The following results are particularly interesting:

$$\begin{aligned} A^k \oplus A^l &\equiv A^{k+l}, \\ (A^k)^l &\equiv A^{kl}, \\ (A \oplus B)^k &\equiv A^k \oplus B^k, \end{aligned}$$

(provided that A^k, B^k have the same defect value), where k, l are positive integers. There is obviously a close resemblance between the above three equations and the *laws of exponents* in number theory; that is, $a^k \cdot a^l = a^{k+l}$, $(a^k)^l = a^{kl}$, $(a \cdot b)^k = a^k \cdot b^k$. This resemblance, we believe, has a far-reaching geometric significance. It also raises related mathematical questions such as: “Is it possible to devise something like a *binomial formula* for shapes?”

We hope to answer some of these questions in the near future.

References

- [1] Altmann, S.L., “*Rotations, Quaternions, and Double Groups*,” Dover, New York, 2005.
- [2] Arbab, F., “RSC: A Calculus of Shapes,” in “*CAD84*,” Butterworth, London, 1984.
- [3] Baylis, W.E. (ed.), “*Clifford (Geometric) Algebras with Applications to Physics, Mathematics, and Engineering*,” Birkhauser-Verlag, Boston, MA, 1996.
- [4] Batten, L.M., “*Combinatorics of Finite Geometries*,” Cambridge University Press, Cambridge, 1986.
- [5] Beckman, F.S., “*Mathematical Foundations of Programming*,” Addison-Wesley, Reading, MA, 1980.
- [6] Bishop, E., “*Foundations of Constructive Analysis*,” McGraw-Hill, New York, 1967.
- [7] Blum, H., “Biological Shape and Visual Sciences, Part I,” *Journal of Theoretical Biology*, **38**, 1973, 205–287.
- [8] Bonnesen, T. and Fenchel, W., “*Theory of Convex Bodies*,” BOS Associates, Moscow, Idaho, 1987.
- [9] Borsuk, K. and Szmielew, W., “*Foundations of Geometry*,” North-Holland, Amsterdam, 1960.
- [10] Bourbaki, N., “*Elements of Mathematics*,” vol. 1: “*Theory of Sets*,” Springer-Verlag, Berlin, 1970.
- [11] Bourbaki, N., “*Elements of Mathematics*,” vol. 2: “*Algebra I*,” Springer-Verlag, Berlin, 1989.
- [12] Bronskill, J. and Venetsanopoulos, A.N., “Multidimensional Shape Description and Recognition Using Mathematical Morphology,” *Journal of Intelligent and Robotic Systems*, **1**, 1988, 117–143.
- [13] Cook, D.J. and Bez, H.E., “*Computer Mathematics*,” Cambridge University Press, Cambridge, 1984.
- [14] Coxeter, H.M.S., “*Regular Polytopes*,” 3rd edn., Dover, New York, 1973.
- [15] Cundy, H.M. and Rollett, A.P., “*Mathematical Models*,” 3rd edn., Tarquin Publications, Norfolk, U.K., 1981.
- [16] Danzer, L., Grünbaum, B. and Kieck, V., “Helly’s Theorem and Its Relatives,” in “*Convexity – Proceedings of the Symposium in Pure Mathematics*,” American Mathematical Society, **VII**, 1963, 101–180.
- [17] Devlin, K., “*The Joy of Sets: Fundamentals of Contemporary Set Theory*,” 2nd edn., Springer, New York, 1994.
- [18] Dyer, C., “Multiscale Image Understanding,” in “*Parallel Computer Vision*” (ed. L. Uhr), Academic Press, Boston, MA, 1987, pp. 171–213.
- [19] Forrest, A.R., “Computational Geometry in Practice,” in “*Fundamental Algorithms for Computer Graphics*” (ed. R.A. Earnshaw), NATO ASI Series F17, Springer-Verlag, Berlin, 1985, pp. 707–724.
- [20] Forrest, A.R., “Computational Geometry and Software Engineering: Towards a Geometric Computing Environment,” in “*Techniques for Computer Graphics*” (eds. R.A. Earnshaw and D.F. Rogers), Springer-Verlag, New York, 1987, pp. 23–37.
- [21] Fu, K.S., “*Syntactic Pattern Recognition and Application*,” Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [22] Fulton, W. and Harris, J., “*Representation Theory*,” Springer, New York, 1991.
- [23] Gardan, Y. and Lucas, M., “*Interactive Graphics in CAD*,” Kogan Page, London, 1984.
- [24] Ghosh, P.K., “*Introducing Interactive Computer Drawing to the Students of Calligraphy and Art with the Help of PaIa TINO System*,” CSI Bangalore Report, April 1982.
- [25] Ghosh, P.K. and Mudur, S.P., “The Brush-Trajectory Approach to Figure Specification: Some Algebraic Solutions,” *ACM Transactions on Graphics*, **3**(2), 1984, 110–134.
- [26] Ghosh, P.K., “*Computational Theoretic Framework for Shape Representation and Analysis Using Minkowski Addition and Decomposition Operators*,” Ph.D. thesis, Tata Institute of Fundamental Research, Bombay (Mumbai), September 1986.

- [27] Ghosh, P.K., "A Mathematical Model for Shape Description Using Minkowski Operators," *Computer Vision, Graphics and Image Processing*, **44**, 1988, 239–269.
- [28] Ghosh, P.K., "A Solution of Polygon Containment, Spatial Planning, and Other Related Problems Using Minkowski Operations," *Computer Vision, Graphics and Image Processing*, **49**, 199, 1–35.
- [29] Ghosh, P.K., "Vision, Geometry, and Minkowski Operators," *Contemporary Mathematics* (American Mathematical Society), **119**, 1991, 63–83.
- [30] Ghosh, P.K., "An Algebra of Polygons through the Notion of Negative Shapes," *CVGIP: Image Understanding*, **54**(1), 1991, 119–144.
- [31] Ghosh, P.K. and Haralick, R.M., "Mathematical Morphological Operations of Boundary-Represented Geometric Objects," *Journal of Mathematical Imaging and Vision*, **6**(2/3), 1996.
- [32] Giardina, C.R. and Dougherty, E.R., "*Morphological Methods in Image Processing and Signal Processing*," Prentice Hall, Englewood Cliffs, NJ, 1988.
- [33] Gilbert, W.J. and Nicholson, W.K., "*Modern Algebra with Applications*," 2nd edn., Wiley-Interscience, 2004.
- [34] Grünbaum, B., "*Convex Polytopes*," Interscience, London, 1967 (2nd edn., Springer, New York, 2003).
- [35] Guibas, L.J., Ramshaw, L. and Stolfi, J., "A Kinetic Framework for Computational Geometry," in "*Proceedings of the IEEE 24th Annual Symposium on Foundations of Computer Science*," 1983, pp. 100–111.
- [36] Guibas, L.J. and Seidel, R., "Computing Convolutions by Reciprocal Search," *Discrete and Computational Geometry*, **2**, 1987, 157–193.
- [37] Haralick, R.M., Sternberg, S.R., and Zhuang, X., "Image Analysis Using Mathematical Morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-9**(4), 1987, 532–550.
- [38] Haralick, R.M. and Shapiro, L.G., "*Computer and Robot Vision*," vol. 1, Addison Wesley, Reading, MA, 1992, Chapter 5.
- [39] Heijmans, H.J.A.M., "*Morphological Image Operators*," Academic Press, Boston, MA, 1994.
- [40] Heyting, A., "*Intuitionism: An Introduction*," 2nd edn., North-Holland, Amsterdam, 1966.
- [41] Hopcroft, J.E. and Ullman, J.D., "*Introduction to Automata Theory, Languages, and Computation*," Addison-Wesley, Reading, MA, 1979 (new edition with a new coauthor, Rajeev Motwani, 2000).
- [42] Horn, K.P. and Weldon, J., "Filtering Closed Curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-8**(5), 1986, 665–668.
- [43] Iwano, K. and Steiglitz, K., "Testing for Cycles in Infinite Graphs with Periodic Structure," in *Proceedings of the ACM Symposium on Theory of Computing*, 1987, pp. 46–55.
- [44] Jantzen, J.C., "*Representations of Algebraic Groups*," Pure and Applied Mathematics 131, Academic Press, Boston, MA, 1987.
- [45] Kallay, M., "Indecomposable Polytopes," *Israel Journal of Mathematics*, **41**, 1982, 235–243.
- [46] Kallay, M., "Decomposability of Polytopes is a Projective Invariant," *Annals of Discrete Mathematics*, **20**, 1984, 191–196.
- [47] Kanungo, T. and Haralick, R.M., "Vector-Space Solution for a Morphological Shape-Decomposition Problem," *Journal of Mathematical Imaging and Vision*, **2**, 1992, 51–82.
- [48] Karl, W.C., Kulkarni, S.R., Verghese, G.C., and Willsky, A.S., "Local Tests for Consistency of Support Hyperplane Data," *Journal of Mathematical Imaging and Vision*, **6**(2/3), 1996, 249–267.
- [49] Kaul, A. and Rossignac, J., "Solid-Interpolating Deformations: Construction and Animation of PIPs," *Computer & Graphics*, **16**, 1992, 107–115.
- [50] Kelly, P.J. and Weiss, M.L., "*Geometry and Convexity*," John Wiley & Sons, Inc., New York, 1979.
- [51] Kim, C.E. and Rosenfeld, A., "Digital Straight Lines and Convexity of Digital Regions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-4**(2), 1982, 149–153.
- [52] Kim, M.-S. and Sugihara, K., "Minkowski Sums of Axis-Parallel Surfaces of Revolution Defined by Slope-Monotone Closed Curves," *IEICE Transactions on Information & Systems*, **E84-D**(11), 2001, 1540–1547.
- [53] Knuth, D.E., "*Digital Press*," American Mathematical Society, Providence, RI, 1981.
- [54] Knuth, D.E., "Mathematical Typography," *Bulletin of the American Mathematical Society*, **1**(2), 1979, 337–372.
- [55] Koenderink, J.J., "*Solid Shape*," The MIT Press, Cambridge, MA, 1990.
- [56] Krause, E.F., "*Taxicab Geometry*," Dover, New York, 1986.
- [57] Kunen, K., "*Set Theory (Studies in Logic and the Foundations of Mathematics)*," North Holland, Amsterdam, 1980.
- [58] Lindeberg, T., "*Scale-Space Theory in Computer Vision*," Kluwer Academic, Dordrecht, 1994.
- [59] Lozano-Perez, T., "Spatial Planning: A Configuration Space Approach," *IEEE Transactions on Computers*, **C-32**(2), 1983, 108–120.

- [60] Lozano-Perez, T. and Wesley, M.A., "An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles," *Communications of the Association for Computing Machinery*, **22**, 1979, 560–570.
- [61] Lucertini, M., Millán Gasca, A., and Nicolo, F., "*Technological Concepts and Mathematical Models in the Evolution of Modern Engineering Systems*," Birkhauser-Verlag, Basel, 2004, p. 89.
- [62] Lyustemik, L.A., "*Convex Figures and Polyhedra*," Dover, New York, 1963.
- [63] Mac Lane, S., "*Algebra*," 3rd edn., American Mathematical Society, Providence, RI, 1999.
- [64] Mandelbrot, B.B., "*The Fractal Geometry of Nature*," W.H. Freeman, San Francisco, 1982.
- [65] Maragos, P., "Pattern Spectrum and Multiscale Shape Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-11**, 1989, 701–716.
- [66] Maragos, P. and Schafer, R.W., "Morphological Skeleton Representation and Coding of Binary Images," *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-34**, 1986, 1228–1244.
- [67] Maragos, P. and Schafer, R.W., "Morphological Filters – Part I: Their Set-Theoretic Analysis and Relations to Linear Shift-Invariant Filters," *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-35**, 1987, 1153–1169.
- [68] Maragos, P. and Schafer, R.W., "Morphological Filters – Part II: Their Relations to Median, Order-Static and Stack Filters," *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-35**, 1987, 1170–1184.
- [69] Matheron, G., "*Random Sets and Integral Geometry*," John Wiley & Sons, Inc., New York, 1975.
- [70] Melder, R.A., "A Survey of Digital Metrics," *Contemporary Mathematics* (American Mathematical Society), **119**, 1991, 95–106.
- [71] Meyer, W., "Indecomposable Polytopes," *Transactions of the American Mathematical Society*, **190**, 1974, 77–86.
- [72] Middleditch, A.F., "The Representation and Manipulation of Convex Polygons," in "*Theoretical Foundations of Computer Graphics and CAD*" (ed. R.A. Earnshaw), Springer-Verlag, Berlin, 1988, pp. 211–253.
- [73] Millman, R.S. and Parker, G.D., "*Geometry: A Metric Approach with Models*," Springer, New York, 1982.
- [74] Minsky, M.L., "*The Society of Mind*," Simon & Schuster, New York, 1987.
- [75] Mount, D. and Silverman, R., "Packing and Covering the Plane with Translates of A Convex Polygon," *Journal of Algorithms*, **11**, 1990, 564–580.
- [76] Mount, D. and Silverman, R., "Combinatorial and Computational Aspects of Minkowski Decompositions," *Contemporary Mathematics* (American Mathematical Society), **119**, 1991, 107–124.
- [77] McMullen, P., "Representations of Polytopes and Polyhedral Sets," *Geometriae Dedicata*, **2**, 1973, 83–94.
- [78] Naur, P. et al., "Revised Report on the Algorithmic Language Algol 60," *The Computer Journal*, **5**, 196, 349–367.
- [79] Pedoe, D., "*The Gentle Art of Mathematics, Circles: A Mathematical View, Geometry and the Visual Arts*," Dover, New York, 1983.
- [80] Pitas, I. and Venetsanopoulos, A.N., "Morphological Shape Decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-12**, 1990, 38–45.
- [81] Preparata, F.P. and Hong, S.J., "Convex Hull of Finite Sets of Points in Two and Three Dimensions," *Communications of the Association for Computing Machinery*, **2**(20), 1977, 87–93.
- [82] Preparata, F.P. and Shamos, M.I., "*Computational Geometry: An Introduction*," Springer, New York, 1985.
- [83] Ray, S., "*Scientific Photography and Applied Imaging*," Focal Press, Oxford, 1999.
- [84] Requicha, A.A.G., "Representations for Rigid Solid Objects," *Computing Surveys*, **12**, 1980, 437–465.
- [85] Requicha, A.A.G. and Rossignac, J.R., "Solid Modeling and Beyond," *IEEE Computer Graphics and Applications*, **12**(5), 1992, 31–44.
- [86] Rogers, D.F. and Adams, J.A., "*Mathematical Elements for Computer Graphics*," McGraw-Hill, New York, 1976.
- [87] Roman, P., "*Some Modern Mathematics for Physicists and Other Outsiders*," vol. 1, Pergamon Press, New York, 1975.
- [88] Rosenfeld, B.A. and Sergeeva, N.D., "*Stereographic Projection*," Mir Publishers, Moscow, 1977.
- [89] Schwartz, J.T., "Finding the Minimum Distance between Two Convex Polygons," *Information Processing Letters*, **13**, 1981, 168–170.
- [90] Serra, J., "*Image Analysis and Mathematical Morphology*," Academic Press, New York, 1982.
- [91] Serra, J., "Introduction to Mathematical Morphology," *Computer Vision, Graphics and Image Processing*, **35**, 1986, 283–305.
- [92] Serra, J. (ed.), "*Image Analysis and Mathematical Morphology*," vol. 2: "*Theoretical Advances*," Academic Press, New York, 1988.
- [93] Shephard, G.C., "Decomposable Convex Polyhedra," *Mathematika*, **10**, 196, 89–95.

- [94] Smilansky, Z., "Decomposability of Polytopes and Polyhedra," *Geometriae Dedicata*, **24**, 1987, 29–49.
- [95] Sugihara, K., Imai, T., and Hataguchi, T., "An Invertible Minkowski Sum of Figures," *Systems and Computers in Japan*, **29**(7), 1998, 33–40.
- [96] Todhunter, I. and Leathem, S.G., "*Spherical Trigonometry*," Macmillan, London, 1949.
- [97] Toussaint, G.I., "Solving Geometric Problems with the Rotating Calipers," in *Proceedings of Melecon '83*, 1983.
- [98] Xu, J., "Decomposition of Convex Polygonal Morphological Structuring Elements into Neighborhood Subsets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(2), 1991, 153–161.
- [99] Yaglom, I.M. and Boltyanskii, V.G., "*Convex Figures*," Holt, Rinehart & Winston, New York, 1961.

Index

- GR*-polygons, 239
- L*-polygons, 237, 238
- R*-polygons, 238, 239
- n* times expansion, 156
- n*-ary relation, 52, 74
- n*-ary composition law, 78
- 2-face theorem, 79, 141, 165

- Abelian group, 95, 100, 114, 119
- additive inverse, 115, 184, 185
- aggregate, 43
- algebra of circles, 36
- algebraic composition law, 77, 78, 79
- algebraic structure, 40, 78, 79, 165
- algebraic system, 79
- analytic function, 75
- analytic geometry, 6, 41, 51
- analytical structure, 40
- antisymmetric, 54, 57
- arithmetization, 6
- associativity, 67, 84, 86, 94
- automorphism, 91, 94

- Backus normal form, 19
- Bezier spline, 29
- bijjective function, 66
- binary composition law, 78
- binary relation, 52, 57
- blends, 29
- boundary addition, 169, 181, 183
- boundary representation, 147, 170, 172, 177
- Bolyai–Labachevskian geometry, 11

- cancellation law, 100, 117
- canonical derivation, 23, 128
- canonical mapping, 73
- canonical projection, 73
- cardinality, 45, 53
- Cartesian coordinate plane, 51
- Cartesian coordinate system, 29, 38
- Cartesian product, 50, 71
- central symmetry, 140
- chain, 58
- class, 43
- closing, 150, 151, 152
- closure, 75
- collection, 43
- combinatorial, 31
- commutative group, 95
- commutative ring, 117
- commutativity, 84, 94, 166
- complete representation, 169, 219
- completeness, 12, 26, 141
- composite function, 66, 70
- composition of transformations, 80
- computational complexity, 36
- concatenation, 20
- conciseness, 9, 33, 143, 149
- congruence class, 115
- congruent transformation, 81, 89
- conjugate of a quaternion, 120
- consistency of axioms, 25
- constraint, 14, 26
- construction rule, 12
- constructive solid geometry, 127

- convex hull, 69, 135, 174, 211
- coordinatization of points, 38
- cover, 55
- cyclic group, 101
- description domain, 15, 20, 26, 141
- diameter, 235
- dihedral angle, 188
- dilation, 129, 150, 167
- distributivity, 84
- division ring, 118
- domain of operation, 27
- duality, 72, 91, 152
- edge length representation, 180
- empty relation, 53
- empty set, 45
- endomorphism, 91
- enumerative scheme, 2
- epimorphism, 88
- equivalence, 55, 72
- equivalence class, 56, 73, 131, 244
- equivalence kernel, 73
- equivalence relation, 72, 166
- erasing, 130, 132, 139, 146
- erosion, 129, 132, 150, 167
- Euclidean group, 105
- Euclidean plane, 51
- explicit axiom, 13, 15
- exponentiation, 81
- external composition law, 78, 81
- facet, 186
- facet representation, 189
- Fermat's conjecture, 216
- field, 119
- finite group, 96
- formal language, 16, 20, 23
- Fourier transform, 29
- free semigroup, 94
- function, 13, 59
- functional analysis, 41
- gap, 235
- generalized convexity, 29, 221
- generalized straight-line segment, 221
- generative scheme, 2
- generator, 82
- genetic method, 13, 79
- geometric and topological property, 15
- geometric dimensionality, 72
- glue operation, 34
- grammar, 18
- graph, 63
- greatest lower bound, 59
- group, 88, 95
- growing, 128
- half-plane, 14
- holomorphic function, 75
- homomorphism, 41, 87, 91
- hyperbolic geometry, 11
- hypercomplex numbers, 218
- icosahedral group, 112
- identity element, 85, 93, 95, 101, 109, 116, 166
- identity function, 70, 102
- identity mapping, 60
- identity relation, 53
- improper rotation, 107
- inconsistent, 10
- indecomposability problem, 215, 242
- independence of axioms, 25
- index set, 44
- infimum, 59
- injection, 65
- integral domain, 118
- internal composition law, 78
- into, 65
- invariant, 218
- inverse element, 86
- inverse function, 67
- invertible, 70, 75, 101
- involution, 88
- isometry, 81, 89, 105
- isomorphic, 71, 87
- isomorphism, 71, 87, 88
- kernel, 34
- language, 94
- least upper bound, 59
- left distributive, 84
- left inverse, 86
- line segment, 46
- linear transformation, 105, 237
- local supporting hyperplane, 195
- lossless description, 26
- lossy image compression, 27, 150
- lower bound, 58

- MAX operation, 174
- mapping, 13
- mathematical morphology, 125, 150
- mathematically complete description, 26
- matrix group, 107
- M-convexity, 222
- measure structure, 40
- mechanical process, 7
- medial axis transform, 157
- member, 44
- metamodel, 6
- metric space, 145
- Minkowski addition, 125, 128
- Minkowski decomposition, 125, 129
- minimality property, 13
- MIN operation, 174
- model, 24
- modulo, 166
- monoid, 93, 115, 134, 165
- monoid homomorphism, 94
- monolithic structure, 37
- monomorphism, 88
- monotone, 220
- morphism, 41, 87
- morphological decomposition, 158, 159
- morphological skeleton, 156
- morphologically indecomposable shape, 216
- multi-scale representation, 32

- natural number, 44
- negative shape, 165, 186
- nonconvex face, 197
- null set, 45
- number system, 115
- numerical algorithm, 31

- octahedral group, 111
- one-to-one, 65
- one-to-one correspondence, 66
- one-to-one onto, 66
- onto, 65
- opening, 150, 152, 158
- orbit, 104
- order of a group, 96
- ordered pair, 50, 71
- orthogonal group, 107
- outer normal, 196

- parametric form, 35
- partial function, 64
- partial order relation, 57
- partial reconstruction, 158
- partially ordered set, 58
- permutation, 70, 90
- permutation group, 96, 97
- perspective projection, 61
- phrase-structured grammar, 18
- physical validity of the shape, 15
- pixel, 148
- Plücker's theory, 71
- point at infinity, 191
- polygonal cover, 219, 221
- power set, 47, 58, 82, 167
- pragmatic approach, 4
- primality test, 216
- prime numbers, 215
- product of transformations, 80
- production rule, 12, 14, 18
- projection, 74
- projection center, 190
- proper rotation, 107
- proper subset, 27, 46, 65, 100, 117
- pure approach, 4
- pure axiom, 12
- purpose-oriented, 8, 25

- quaternion, 119
- quotient set, 57

- range of operation, 27, 65
- rational B-spline, 29
- realization, 92, 101, 184, 192, 199, 207, 230
- redundancy, 33
- reflective symmetry, 89
- reflexive, 53
- regular solid, 108
- regularized intersection, 28
- relation, 52
- relative supporting
 - hyperplane, 195
- representative element, 57
- restriction, 14
- right distributive, 84
- right inverse, 86
- ring, 113
- rings with identity, 117
- rings without divisors of zero, 117
- rotation, 101, 122
- rotational symmetry, 89, 139
- rules, 7

- scalar multiplication, 82
- self-crossing, 30
- self-crossing polygon, 182, 200, 205, 209
- semantic ambiguity, 22
- semantics, 21
- semigroup, 93
- set complement, 48
- set difference, 48
- set intersection, 48
- set union, 47
- shape operator, 14
- simple decomposition, 161
- simply connected shape, 16
- singleton, 45
- skeleton element, 157
- skeleton transform, 157
- skeletoning, 157
- slope diagram, 182
- space–complexity tradeoff, 34
- spatial occupancy, 35, 148
- special orthogonal group, 107
- sphere-geometry, 36
- stabilizer, 104
- stereographic projection, 191
- strong decomposition, 218
- structuring element, 129
- subsystem, 83
- successor function, 13
- supporting function, 169, 172
- supporting function vector, 178
- supporting hyperplane, 169, 195
- supremum, 59
- surjection, 65
- swept volume, 22
- symmetric difference, 48
- symmetry, 53, 70, 89, 100
- symmetry group, 104
- syntactic ambiguity, 22
- syntax, 21
- taxicab convex class, 220
- taxicab line segment, 221, 242
- ternary relation, 25
- tetrahedral group, 110
- thinning, 151
- thickening, 151
- three-dimensional geometric
 - space, 46
- topological space, 145
- topological structure, 40
- topologically equivalent, 72
- total function, 64
- totally ordered set, 58
- Toy system, 14, 26
- transformation, 60
- transformation group, 102
- transitive, 54, 57
- translational symmetry, 140
- trial division, 216
- triangle theorem, 140
- umbra, 155
- unary composition law, 78
- unary operation, 47, 75, 182
- unified algorithm, 205
- uniqueness in description, 15, 22
- unit element, 85
- unitary divisor, 216
- universal approximating
 - class, 217
- universal relation, 53
- universal set, 43
- universal Turing machine, 6
- universe of discourse, 43
- upper bound, 58
- utilitarian approach, 4
- Venn diagrams, 49
- vocabulary, 7
- voxel, 148
- weakly taxicab convex
 - polygon, 222
- WTC domain, 226, 234, 236, 242